



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola d'Enginyeria de Telecomunicació
i Aeroespacial de Castelldefels

TREBALL DE FI DE GRAU

TÍTOL DEL TFG: Millora de les funcionalitats d'una eina per a la gamificació

TITULACIÓ: Grau en Enginyeria d'Aeronavegació

AUTORS: Sergi Nogués Farrés
César Flores Navarro
Lovejinder Singh

DIRECTORS: Miguel Valero
Roc Mesenguer

DATA: 1 de febrer de 2019

Títol: Millora de les funcionalitats d'una eina per a la gamificació

Autors: Sergi Nogués Farrés
César Flores Navarro
Lovejinder Singh

Directors: Miguel Valero
Roc Mesenguer

Data: 1 de febrer de 2019

Resum

La realització d'aquest treball s'ha centrat en la millora i ampliació d'un projecte ja existent basat en la programació d'una eina que permeti incorporar la gamificació en entorns educatius (aplicació anomenada **Classpip**). La gamificació és la utilització de diverses tècniques, elements i dinàmiques dels jocs en activitats no lúdiques com ara l'educació, com per exemple mitjançant l'assignació de punts i recompenses als alumnes per tal de motivar-los a realitzar les seves tasques correctament. Per tal de dur a terme les esmentades millores, s'ha continuat treballant amb l'entorn de programari definit en les anteriors versions d'aquest projecte, el qual va ser iniciat per diversos estudiants de l'EETAC conjuntament amb el departament d'Arquitectura de Computadors (AC).

Partint de l'anterior versió d'aquesta eina, es poden diferenciar diversos aspectes importants en els quals s'ha centrat el nostre projecte per tal de desenvolupar millores, tant en l'aplicació com en l'entorn de documentació que l'envolta. Per una banda, s'ha treballat en la millora, optimització i organització dels mòduls ja existents, és a dir, s'han afegit noves funcionalitats als mòduls ja existents i s'ha reorganitzat el codi per tal de fer-lo més eficient i robust. A més a més, ha estat primordial realitzar les **connexions** entre aquests mòduls, havent partit de dos projectes separats amb mòduls diferenciats i unint-los fent possibles les seves funcionalitats conjuntament. D'aquesta manera s'ha aconseguit transformar un conjunt de mòduls amb cap interrelació en una eina per a la gamificació en l'educació. També s'han creat nous mòduls partint de zero i s'han implementat en el conjunt de programari de l'aplicació. Per últim, s'ha realitzat una actualització dels materials per a l'onboarding (el conjunt de materials d'aprenentatge sobre el projecte i les seves eines, destinat als nous programadors d'aquest), s'ha dut a terme la preparació de diferents vídeo-tutorials per als professors que vulguin utilitzar Classpip (projecció externa) i s'han identificat una sèrie de futures millores per a properes versions del projecte.

Per tal de poder fer possible el desenvolupament dels propòsits esmentats en el paràgraf anterior, s'ha seguit una dinàmica de treball molt organitzada per tal que els diferents integrants poguéssim coordinar la nostra feina simultàniament. Primerament, es va partir del material d'onboarding ja existent i, seguint els tutorials, es va realitzar una familiarització amb el codi de l'aplicació, l'entorn de treball i el programari utilitzat. Així doncs, es va decidir utilitzar la eina **GIT** per tal de poder desenvolupar diferents versions de codi de forma simultània i poder ajuntar-les de manera funcional un cop acabada cada millora de l'aplicació. Un cop entès l'entorn de treball i la seva dinàmica, es va optar per primer realitzar la fusió entre diferents mòduls independents (imprescindible per tal de poder continuar el desenvolupament de Classpip) per més tard entrar en la creació de nous mòduls i funcionalitats extres.

Resum

L'objectiu principal d'aquest treball ha estat, en tot moment, transformar un conjunt de mòduls independents programats en un mateix entorn de codi en una versió vendible de l'aplicació per tal de poder fer-hi proves i preparar-la per a futures actualitzacions. D'altra banda, també s'ha centrat en la millora d'uns tutorials per a nous programadors de l'eina amb carències d'explicacions detallades i poca claredat que entorpien el progrés del desenvolupament amb els nous membres. Per últim el conjunt de documentació del projecte ara compta amb vídeo-tutorials actualitzats per a professors segons la modalitat amb la que vulguin jugar a Classpip, incloent totes les noves funcionalitats implementades en la darrera versió. Un cop acabat el projecte es pot concloure que aquestes millores han estat dutes a terme satisfactòriament, ja que s'han implementat totes les possibles a l'aplicació amb el temps del que es disposava i s'ha extès el material d'Onboarding notablement per tal de fer l'eina més amigable tant per a futurs desenvolupadors com per a professors usuaris d'aquesta.

Title : Improvement of a gamification tool functionalities

Authors: Sergi Nogués Farrés
César Flores Navarro
Lovejinder Singh

Advisors: Miguel Valero
Roc Mesenguer

Date: February 1, 2019

Overview

The development of this project has focused on the improvement and extension of an already existing one, based in the programming of a tool that allows to incorporate gamification in educational environments (the application is named **Classpip**). Gamification consists in using various game techniques, elements and dynamics in non-playful activities such as education, for example by using a point assignment system and rewards for the students in order to motivate them to perform their tasks correctly. In order to carry out these improvements, the same programming environment of previous versions has been kept. This project was started by several EETAC students, in collaboration with the “Arquitectura de Computadors” (AC) department.

Taking from the last version of this tool, various important aspects can be observed in which our project has been centered in order to develop improvements, not only in the application but also in the documentation it involves. On the one hand, work has been made on the improvement, optimization and organization of the already existing modules, in other words, new functionalities have been added to the already existing modules and the code has been reorganized in order to make it more efficient and more robust. Moreover, it has been overriding to create the **connections** between these modules, starting from two separate projects with differentiated modules and mixing them, making their functionalities possible in a combined way. This way, it has been achieved to transform a whole set of modules with no relation between them in a gamification tool with educational purposes. Some new modules have also been created from scratch and they have been implemented into the set of the application software. Finally, an upgrading of the onboarding materials has been carried out (the set of materials that allows the new members of the project to learn about it and the tools used on it), in addition to the preparation of several video-tutorials for the teachers willing to use Classpip (external projection) and, moreover, a series of improvements have been identified for future versions of this project.

In order to make the development of the previously stated objectives possible, a very organized working dynamic has been followed to allow the different members of this project to coordinate their work simultaneously. First of all, starting from the already existing Onboarding materials and following the tutorials, a familiarization with the application code, the working environment and the used software were carried out. Therefore, it was decided to use the tool **GIT** in order to be able to develop several code versions simultaneously and later on, mix them together in a functional way once each application improvement was finished. Once the working environment and its dynamic were understood, it was chosen to merge the different and independent modules first (something that was indispensable in order to be able to continue with the development of the Classpip application), followed by the creation of new modules and the design of extra functionalities.

Overview

The main objective of this project has been, at all times, to transform a whole set of independent modules programmed in the same code environment into a salable version of the application in order to be able to make tests on it and prepare it for future updatings. On the other hand, it has also been centered in the improvement of some tutorials for new programmers of this tool, which lacked of detailed explanations and had poor clarity, things that obstructed the progress of the application development in hands of new members. At last, the documentation of the project now includes some updated video-tutorials for professors, depending on the mode they want to play Classpip with, including all the new upgradings implemented in the last version of the tool. Once the project has been finished, we can conclude that these improvements have been carried out satisfactorily, since all the possible ones have been implemented within the available time. Moreover, the Onboarding material has been notably extended in order to make it a more friendly tool, not only for future developers but also for the teachers that will be its users.

Aquest treball va dedicat
a totes les persones que
han fet possible que en
pocs dies puguem dir
que som **enginyers**

ÍNDEX

Introducció	1
CAPÍTOL 1. LA GAMIFICACIÓ	3
1.1. Què és la Gamificació?	3
1.2. Exemples de Gamificació en educació	4
1.2.1. Exemple 1: ClassCraft	4
1.2.2. Exemple 2: Class Dojo	5
1.2.3. Exemple 3: Show My Homework	6
CAPÍTOL 2. INICIACIÓ AL PROJECTE	9
2.1. Motivació	9
2.2. Arquitectura del projecte	9
2.3. Descripció de l'aplicació	10
2.3.1. Grups	12
2.3.2. Punts i insígnies	12
2.3.3. Col·leccions	12
2.3.4. Qüestionaris	12
2.3.5. Competicions	12
2.3.6. Equips	13
2.3.7. Assistència	13
2.4. Objectius	14
CAPÍTOL 3. MATERIALS PER ONBOARDING	17
3.1. Millores en el manual d'instal·lació de l'entorn de desenvolupament	17
3.2. Passeig pel projecte	18
3.2.1. Posada en marxa de l'aplicació	18
3.2.2. Canvi en una pàgina html	18
3.2.3. Canvi en una funció	19
3.3. Millores tutorial Mesa	19
3.4. Treball en equip amb el git	20

3.4.1. Els repositoris de Classpip	20
3.4.2. Git en equip: metodologia i resultats	22
3.4.3. Posada en marxa de la darrera versió de Classpip	23
3.5. Ampliació catàleg d'errors	24
 CAPÍTOL 4. MILLORA, CREACIÓ I CONNEXIÓ DE MÒDULS	 25
4.1. Unió dels dos projectes inicials	26
4.2. Mòdul de Punts i Insígnies	27
4.2.1. Actualització del model de punts	28
4.2.2. Connexió amb el mòdul d'Equips i millora de la interfície	29
4.2.3. Classificació dels estudiants	30
4.3. Mòdul de Col·leccions	32
4.3.1. Intercanvi de cromos	34
4.4. Mòdul de Competicions	35
4.4.1. Premis assignables manualment	37
4.4.2. Premis automatitzables	39
4.4.3. Resultats Jornada a partir del nombre de Punts del participant	42
4.5. Mòdul d'Equips	43
4.5.1. Visualització d'equips	43
4.6. Mòdul d'Assistència	45
 CAPÍTOL 5. PROJECCIÓ EXTERNA	 47
5.1. Com jugar amb assignació de punts	47
5.2. Com jugar a una competició	47
5.3. Com crear equips i jugar amb ells	47
 Conclusions	 49
 Bibliografia	 55
 APÈNDIX A. Manual d'instal·lació de l'entorn de desenvolupa- ment	 59
A.1. Dependències	59
A.1.1. Instal·lació del NODE i NPM	59

A.1.2. Instal·lació del GIT	60
A.1.3. Instal·lació del SOURCETREE	61
A.1.4. Instal·lació del VISUAL STUDIO CODE	61
A.1.5. Instal·lació de l'ANDROID STUDIO	62
A.2. Repositoris	62
A.2.1. Classpip-services	63
A.2.2. Classpip-mobile	64
A.2.3. Classpip-dashboard	66
APÈNDIX B. Tutorial Mesa	67
B.1. Repositoris de partida	67
B.2. Objectiu i definició de l'objecte	69
B.3. Passos a seguir per dur a terme el tutorial	71
B.3.1. Creació del model 'Mesa' i les dades al repositori Services	71
B.3.2. Visualització a l'API	74
B.3.3. Creació de les dades inicials	76
B.3.4. Verificació de les dades mitjançant Postman i l'API (pas opcional)	79
B.3.5. Creació de les pàgines al repositori Mobile	80
B.3.6. Creació de les pàgines al repositori Dashboard	90
APÈNDIX C. Manual de GIT	95
C.1. Primers passos amb el repositori Classpip	95
C.2. Protocol per treballar en grup en Git	96
C.3. Exemple de treball amb el Git: Posada en marxa de la darrera versió de Classpip	99
APÈNDIX D. Catàleg d'errors	101
D.1. Errors a Windows	101
D.1.1. Error 1: Can't find Python executable:	101
D.1.2. Error 2: Unable to clone Git repository due to self signed certificate:	103
D.1.3. Error 3: Error de connexió amb la base de dades:	103
D.2. Errors a Ubuntu	104
D.2.1. Error 1: Permission denied:	104
D.2.2. Error 2: Bucle de l'StrongLoop:	105
D.2.3. Error 3: Not found: make:	106
D.2.4. Error 4: Plugins no trobats:	107

D.3. Errors generals	109
D.3.1. Error 1: Error Visual Package 8.1 SDK:	109
D.3.2. Error 2: Error Visual Package 8.1 SDK:	110
D.3.3. Error 3: Error intern del servidor:	111

ÍNDEX DE FIGURES

1.1 La interfície de Show My Homework	6
2.1 Esquema de l'arquitectura de l'aplicació	10
2.2 Projectes realitzats prèviament	11
3.1 Branques Dashboard Classpip	21
4.1 Aparença inicial de la interfície del mòdul de Punts i Insígnies	27
4.2 Model Punts i Insígnies: point.json	28
4.3 Model Punts i Insígnies nova interfície	29
4.4 Punts de l'alumne	30
4.5 Punts de l'alumne	30
4.6 Punts dels equips	31
4.7 Insígnies de l'alumne	31
4.8 Pàgina principal de col·leccions	32
4.9 Pàgina d'una col·lecció específica	33
4.10 Finestra cromó seleccionat	34
4.11 Punts del tipus Competició	36
4.12 Premis tipus Lliga	37
4.13 Premis tipus Torneig	38
4.14 Premis tipus Cromó	38
4.15 Menú de creació de Lliga	39
4.16 Menú de creació de Torneig	39
4.17 Pàgina d'automatitzacions Lliga	40
4.18 Pàgina d'automatitzacions Torneig	40
4.19 Pàgina d'automatitzacions Lliga	41
4.20 Pàgina d'automatitzacions Torneig	41
4.21 Opció de decidir la jornada pels punts en la classificació general	42
4.22 Menú del mòdul d'Equips	43
4.23 Visualitzar Equips	44
4.24 Pàgina de visualització de l'equip	44
4.25 Menú del mòdul d'Assistència	45
4.26 Menú del mòdul d'Assistència	45
A.1 Llista d'instal·ladors del node	59
B.1 Visualització dels punts de menú del resultat final del manual	69
B.2 Visualització de la pàgina de menú del resultat final del manual de la versió mobile	70
B.3 Visualització de la pàgina de menú del resultat final del manual de la versió dashboard	70
B.4 mesa.js	71
B.5 mesa.json	72
B.6 student.json	73

B.7	config.json	74
B.8	Visualització de l'objecte a l'API	74
B.9	Visualització de les funcionalitats de l'API	75
B.10	Visualització de les funcions creades en l'objecte student	75
B.11	Nou arxiu 01-01-create-mesa.js	76
B.12	Carpeta d'assets del repositori Mobile	77
B.13	Arxiu 06-create-students.js	78
B.14	Visualització del token en el perfil	79
B.15	Lloc on introduir el token al Postman	80
B.16	Lloc on introduir el token a StrongLoop (l'API)	80
B.17	Visualització dels tres arxius del directori de la pàgina 'mesa'	81
B.18	Classe del model Mesa creada al nou arxiu <i>mesa.ts</i> en el directori especificat	81
B.19	Modificació del model "Student" a <i>student.ts</i>	82
B.20	Nou arxiu provider anomenat <i>mesa.service.ts</i> creat al directori especificat	82
B.21	Mètode 'get mesa' de l'API	83
B.22	Definició de la ruta dels mètodes de l'API de l'objecte mesa	83
B.23	Esquema explicatiu de l'accés a la pàgina de mesa	84
B.24	Punt d'entrada a la MesaPage des de MenuPage a l'arxiu <i>menu.ts</i>	84
B.25	Mètode que conté el direccionament a la MesaPage (també a <i>menu.ts</i>)	85
B.26	Botó que crida al mètode de direccionament a la MesaPage	85
B.27	Visualització del botó de direccionament	85
B.28	Mètode <i>getter</i> de la taula de l'estudiant	86
B.29	Botó de direccionament a la MesaPage en el menú Home	86
B.30	Fitxers de la pàgina de la taula	87
B.31	Nou arxiu <i>mesa.ts</i>	87
B.32	<i>imports</i> necessaris a l'arxiu <i>mesa.ts</i>	88
B.33	Nou arxiu <i>mesa.scss</i>	88
B.34	Nou arxiu <i>mesa.html</i>	89
B.35	Etiquetes de traducció dels labels a l'arxiu <i>es.json</i>	89
B.36	Definició de les URLs per al Dashboard	90
B.37	Exportació del servei Mesa	90
B.38	Definició del component a l'arxiu <i>mesa.ts</i>	91
B.39	Resta del nou arxiu <i>mesa.ts</i>	91
B.40	Nou arxiu <i>mesa.scss</i>	92
B.41	Nou arxiu <i>mesa.html</i>	92
B.42	Botó de direccionament a la pàgina Mesa	93
B.43	Definició del path de la pàgina Mesa	93
C.1	Exemple de línia de treball	98
D.1	Error can't find Python executable	101
D.2	Error can't find Python executable	102
D.3	Connexió amb la dB a <i>datasources.json</i>	103
D.4	Error permission denied	104
D.5	Error del bucle de l'StrongLoop	105
D.6	Error not found: make	106
D.7	Informació dels plugins no trobats	107
D.8	Plugin que es queda sense identificar	108

D.9 Panell de control de l'ordinador	109
D.10Error de Visual Package 8.1 SDK	110
D.11Missatge d'error intern en el servidor	111

ÍNDIX DE TAULES

4.1 Millores realitzades al projecte	25
--	----

INTRODUCCIÓ

Avui en dia els professors de qualsevol nivell educatiu, ja sigui d'educació primària, secundària o bé d'educació superior, s'enfronten a una manca de motivació i interès per part dels alumnes en gran part degut a l'aparició de les noves tecnologies, com per exemple els videojocs. Una solució és emprar les noves tecnologies com a eina de motivació i aprenentatge.

Aquest projecte es basa en la creació d'una eina per a la **Gamificació** que utilitza les mecàniques que ens ofereixen els jocs en general en l'àmbit educatiu per tal d'obtenir millors resultats en els alumnes. Tota la informació relacionada amb la Gamificació en l'educació es troba explicada en el **Capítol 1**, on s'aprofundeix més en el terme de gamificació i després es mostren 3 exemples pràctics de com diferents eines de gamificació han resultat útils a l'hora d'aplicar-se a l'educació.

L'aplicació de **Classpip** és una eina per a la Gamificació que ha estat desenvolupada per diferents alumnes pertanyents a la EETAC amb la col·laboració del departament d'Arquitectura de Computadors (AC). Aquesta eina, que es troba disponible tant per al dispositiu mòbil com per a l'ordinador, està composta per un conjunt de mòduls que el professor pot utilitzar de forma individual o conjunta. La unió, millora i connexió dels mòduls ja existents així com la creació de nous mòduls, forma part dels objectius d'aquest projecte explicats més endavant. Els mòduls que la componen són els següents:

- **Grups:** El professor pot gestionar els alumnes que té a cada assignatura que imparteix.
- **Assistència:** Permet assignar punts als alumnes que arriben puntuals a classe.
- **Punts i Insígnies:** Permet al professor assignar punts i insígnies als alumnes com a recompensa o premi.
- **Qüestionaris:** Permet avaluar un estudiant a partir de la creació de qüestionaris.
- **Col·leccions:** Permet al professor assignar cromos als alumnes com a recompensa o premi. Els cromos pertanyen a una col·lecció assignada prèviament a un grup d'alumnes d'una assignatura. L'objectiu de l'alumne és completar la col·lecció de cromos.
- **Competicions:** Permet al professor crear competicions entre els alumnes, ja sigui del tipus **Lliga** o del tipus **Torneig de tennis**.
- **Equips:** Permet al professor crear equips d'alumnes dins d'un grup de classe.

Amb l'objectiu de facilitar la feina als futurs programadors d'aquest projecte s'han creat un conjunt de manuals i tutorials anomenats **Materials d'Onboarding**. Les millores aportades a aquest material d'aprenentatge es troben descrites al **Capítol 3**. Tot el material d'onboarding, amb les millores incloses en aquest projecte, es troba en l'**Apèndix**.

Un dels principals objectius d'aquest projecte ha estat millorar l'aplicació de Classpip, unificant els mòduls existents, afegint noves funcionalitats i creant nous mòduls. El **Capítol 4**

recull totes les millores que s'han dut a terme i s'hi explica amb detall el seu funcionament, juntament amb una descripció general de l'aplicació.

Per últim, el **Capítol 5** es centra en la **Projecció Externa**, és a dir, material d'ajut als professors per poder utilitzar aquesta aplicació. Amb aquest objectiu s'han creat videotutorials dirigits als professors que expliquen tres maneres diferents d'utilitzar l'aplicació de Classpip. Una manera és jugant únicament amb el mòdul de Punts i Insígnies, una altra és només crear Competicions i una tercera és crear equips i jugar amb ells. La idea principal és que els professors aprenguin a utilitzar l'aplicació de forma senzilla i entenguin que no es necessari utilitzar tots els mòduls alhora.

CAPÍTOL 1. LA GAMIFICACIÓ

Aquest projecte es basa en el desenvolupament d'una eina per a la gamificació, però, que és la **Gamificació** i com s'aplica en l'educació?

En aquest primer capítol s'introduirà el terme de gamificació i es mostraran diversos exemples diferents de com aquesta s'ha aplicat en l'educació.

1.1. Què és la Gamificació?

La gamificació és la utilització de les tècniques, elements i dinàmiques dels jocs en activitats no lúdiques amb la finalitat de potenciar la motivació, absorbir millor alguns coneixements, millorar alguna habilitat o recompensar accions concretes, entre d'altres objectius..

Gamificar és un concepte que neix de l'aprenentatge que proporcionen els jocs, les seves mecàniques, eines, desenvolupaments, afrontaments i la manera en la que les persones són satisfetes a través de la oferta de recompenses i/o estímuls per tasques desenvolupades de una manera o un altre en àmbits no relacionats necessàriament amb el joc. En definitiva, l'objectiu és aprendre a canvi de diversió.

En educació està guanyant terreny degut al seu caràcter lúdic, que facilita la interiorització de coneixements d'una manera més divertida. El model de joc funciona perquè aconsegueix motivar als alumnes, augmentant el compromís d'aquests i incentivant l'ànim de superació.

En la gamificació s'utilitzen tècniques extrapolades dels jocs, com per exemple:

- Recompenses en funció dels objectius aconseguits
 - Acumulació de punts
 - Augment de nivells
 - Obtenció de premis o regals
- Classificacions
- Desafiaments
- Reptes

Utilitzar gamificació a les aules és eficaç sempre i quan s'utilitzi per animar als estudiants a progressar a través dels continguts de aprenentatge, per influir en el seu comportament o accions i per generar motivació. És possible motivar als alumnes amb la introducció de una metodologia que inclogui reptes, metes, etc. Aquests elements fomenten la participació o l'acció en els éssers humans en general. A continuació s'expliquen tres exemples pràctics on s'han aplicat algunes de les tècniques esmentades amunt i en els quals la gamificació ha resultat útil en l'educació.

1.2. Exemples de Gamificació en educació

1.2.1. Exemple 1: ClassCraft

Classcraft és una eina per a la gamificació en l'educació creada per Shawn Young (d'origen Canadenc) el 2013. Es tracta d'una aplicació web que permet als professors dirigir un joc de rol en el que els seus alumnes encarnen diferents personatges, amb la finalitat de fomentar el treball en equip, augmentar la motivació dels estudiants i afavorir un millor comportament a classe.

Un joc d'interpretació de rols, és un joc interpretatiu i narratiu en el que els jugadors assumeixen el "rol" de personatges imaginaris dins d'una història o trama, interpreten els seus diàlegs i descriuen les seves accions.

A la pràctica, l'aplicació funciona de la següent manera:

- Els alumnes es tornen guerrers, mags o curanderos i tenen poders èpics.
- Els poders donen privilegis dins de la classe com per exemple el poder Caça que permet a l'alumne menjar a classe; o el poder Contra atac que permet a l'alumne tenir una pista en una pregunta d'examen. Els poders poden proporcionar qualsevol cosa que serveixi de motivació a l'alumne.
- Per aconseguir poders els alumnes han de guanyar punts d'experiència (XP) que s'aconsegueixen amb el bon comportament. Un exemple de bon comportament és contestar correctament a una pregunta o ajudar a un altre alumne amb la feina.
- Els alumnes també poden perdre punts de salut (HP) per mal comportament. Si un alumne perd tots els punts de salut rebrà una conseqüència aleatòria.
- Els alumnes juguen en equips i per tant si un alumne perd tots els HP la resta de l'equip també surt perjudicat.

El Juliol del 2014, 7000 professors de 50 països diferents ja havien adoptat l'aplicació Classcraft com a complement per a les seves classes. El diari "La actualidad" informava la tardor del 2014 que 1000 classificats de tot el món utilitzaven aquest mètode. Quan el joc es va posar en línia, la tendència sobre Reddit al 2014 indicava 130 000 visitants per dia. Actualment podem instal·lar de forma gratuïta al nostre mòbil l'aplicació de Classcraft al Google Play d'Android. Té una puntuació de 4,6 (5000 opinions) i més de 100 000 descàrregues. L'aplicació és compatible amb Chrome, Android, Windows, IOS i OS X.

1.2.2. Exemple 2: Class Dojo

Class Dojo és una aplicació que permet als mestres de les escoles crear una xarxa de comunicació a dins de la classe, fent alhora que els alumnes aprenguin d'una manera divertida i més atractiva. Bàsicament es basa en, mitjançant un programa d'**assignació de punts** (una de les tècniques que s'han esmentat prèviament) usat pel mestre, poder recompensar els seus alumnes si realitzen de forma satisfactòria les tasques encomanades a classe. Si bé és cert que aquesta aplicació està dissenyada per a estudiants en cursos menys avançats (nens de primària), és un exemple perfecte de gamificació en l'ensenyament i les seves funcionalitats es podrien extreure i extrapolar a cursos superiors com secundària o més enllà.

Aquesta eina de la gamificació, creada l'any 2011 per un grup d'emprenedors del Regne Unit, va ser utilitzada per més de 12000 professors durant les cinc primeres setmanes des del seu llançament. Actualment està disponible en més de 35 idiomes i ha sigut guardonada amb nombrosos premis de l'àmbit de l'educació al llarg dels anys. El fet que aquesta aplicació hagi tingut tant d'èxit es deu a que el professor pot crear una comunitat virtual dins de la seva classe per tal d'interconnectar no tan sols els propis alumnes si no també els mestres i els alumnes amb els pares o tutors d'aquests. D'aquesta manera, Class Dojo pot ser utilitzada tant per estudiants com per mestres, pares i fins i tot el cap de l'escola. Els alumnes poden compartir en carpetes que el professor haurà creat allò que han après a classe i els altres ho poden veure i revisar. A més a més, es poden enviar missatges i fotografies als pares o tutors, podent mantenir un contacte quan sigui necessari amb el mestre per realitzar les tutories i tenir un feedback positiu i immediat sobre el desenvolupament dels alumnes. Aquests missatges poden ser traduïts de forma automàtica a més de 30 idiomes gràcies al sistema de funcionalitats implementades dins de l'aplicació. Fins i tot és possible per als pares o tutors veure en temps real què està passant a la classe i quina dinàmica s'està duent a terme.

Altres funcionalitats d'aquesta aplicació és la capacitat del mestre per crear grups d'alumnes dins de la seva classe i assignar tasques a cada un, rebent recompenses si la feina ha estat ben feta. També s'inclouen explicacions en forma de vídeo que fan que l'aprenentatge per als alumnes sigui més visual i entretingut en certes ocasions. Altres opcions per al mestre són les de deixar veure als alumnes el temps restant que tenen per entregar una tasca, en què haurien de ser treballant en cada moment de la classe, poder posar una música adequada per a l'ambient de treball... Pel que fa als alumnes, cada un pot escollir un avatar que es mostrarà a la resta de companys i rebre recompenses per a millorar el seu perfil segons es van realitzant tasques de manera adequada.

Tal i com s'explica al link situat més avall, una professora d'institut dels Estats Units documenta el seu ús de ClassDojo, explicant que gràcies a que queda constància en l'aplicació de les notes dels seus alumnes i les entregues de classe, pot recollir dades sobre el comportament d'aquests (organització, hàbits de treball, punts forts i febles...) i adaptar les seves classes de manera dinàmica durant el curs segons les necessitats:

<https://www.commonsense.org/education/website/classdojo-teacher-review/3907316>

D'aquesta manera, procura premiar els estudiants amb punts i recompenses quan fan les coses bé per encoratjar-los a seguir endavant i dur els estudis del curs al dia. A més a més resulta una eina de seguiment molt poderosa a l'hora d'apropar-se als seus alumnes, ja que permet recollir informació detallada de cada un i oferir ajuda per enfortir les assignatures amb pitjors notes.

Actualment podem instal·lar aquesta aplicació gratuïtament al nostre dispositiu mòbil buscant-la al Google Play d'Android, tot i que també es troba disponible per a iOS i qualsevol navegador web. Al Google Play té una puntuació de 4,4 (166000 opinions) i més de 10 milions de descàrregues.

1.2.3. Exemple 3: Show My Homework

Show My Work és un aplicació que connecta el professor amb els seus alumnes, i al mateix temps, aquests alumnes amb els seus pares. D'aquesta manera, el professor avalua els progressos dels alumnes organitzant activitats com qüestionaris o deures sobre el contingut de les seves classes. Aquesta aplicació disposa d'un calendari en el que poden visualitzar tots els esdeveniments de la classe (excursions, deures, exàmens, etc.). Disposa també d'un fòrum on tant els alumnes com els professors i l'escola poden interactuar entre ells. Per altra banda, l'aplicació permet als pares fer un seguiment sobre els seus fills. Poden saber en qualsevol moment si els seus fills estan lliurant els deures i les notes que estan traient. I també permet detectar les àrees en el que l'alumne ha de millorar.

A la pràctica, el professor organitza cada setmana diferents reptes i concursos per cada una de les seves classes. Els alumnes participen en aquests esdeveniments, que poden ser cues i el professor rep els resultats. Aquests resultats també són visibles pels pares dels alumnes. També li permet al professor pujar explicacions en format de vídeo que l'alumne pot consultar en qualsevol moment. Treballa amb un sistema de insígnies i punts que recompensen els progressos de l'alumne. A la figura 1.1(a) i 1.1(a) es poden observar captures de diferents funcions que ofereix l'aplicació.

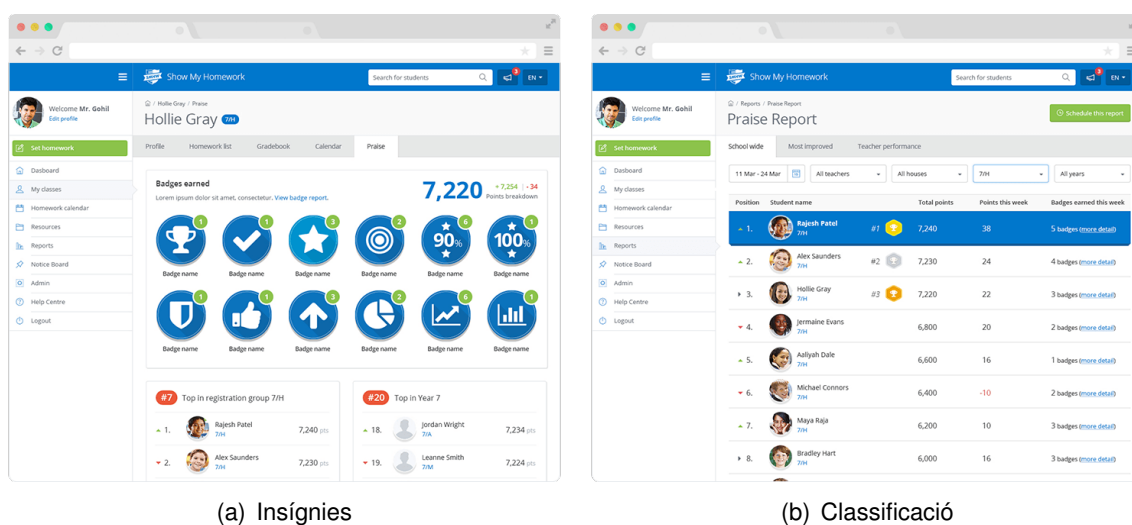


Figura 1.1: La interfície de Show My Homework

En aquestes figures es mostren com Show My Homework posa en pràctica dues de les

tècniques de gamificació:

- Insígnies: premis obtinguts pels progressos fets en classe.
- Classificacions: motivació per competir entre els alumnes i superar-se entre ells.

Actualment, segons la pròpia aplicació, té més d'un milió d'usuaris actius mensualment i està sent utilitzada en més de 1500 escoles globalment. Principalment es va llançar a Regne Unit i només està disponible en anglès. Es pot sol·licitar una versió 'demo' per provar la implementació del programa en una escola. En acabar el període de proves, s'ha de contractar el servei. Actualment s'utilitza en diversos centres de Regne Unit.

<https://teamsatchel.wistia.com/medias/37t1f8a6cj?wtime=0>

<https://www.educationalappstore.com/app/show-my-homework>

Podem trobar l'aplicació tant a iOS com a Android, on té més de 500.000 de descàrregues i una puntuació de 4,1. És totalment gratuït, encara que per utilitzar-la, es necessita la contractació del servei complet.

A continuació, en el Capítol 2, es passa a descriure en termes genèrics quina es l'estructura tant del projecte com de l'aplicació, a més a més dels objectius que es volen assolir amb la realització d'aquest treball.

CAPÍTOL 2. INICIACIÓ AL PROJECTE

Classpip és una eina per a la Gamificació en l'educació que ha estat desenvolupada per estudiants pertanyents a la EETAC. El projecte el va començar el Ferran amb la creació de l'arquitectura de l'aplicació i, més endavant, Guillermo, Manuel, Daniel, la Maria del Mar, l'Anna i el Joan van desenvolupar i millorar un conjunt de mòduls.

En aquest capítol primer s'expliquen els motius que han dut al desenvolupament d'aquest projecte, després una descripció general d'aquest i de l'aplicació i, finalment, els principals objectius que s'han dut a terme.

2.1. Motivació

L'aplicació de Classpip és una plataforma de software creada el 2016 per part del departament d'Arquitectura de Computadors de la Universitat Politècnica de Catalunya, que ha estat desenvolupada per alumnes de la Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels (EETAC). Aquest projecte té com a motivació unir tots els mòduls ja existents d'aquesta plataforma amb la finalitat de crear una versió millorada que implementi funcionalitats conjuntes de tots els mòduls existents, a més a més de crear-ne de nous i aplicar funcionalitats extres. Per altra banda, serà també prioritari expandir i millorar la documentació que envolta el projecte, és a dir, el material d'Onboarding (que recull tutorials d'instal·lació i introducció al projecte per a nous membres del projecte) i la projecció externa (consistint en la preparació d'una sèrie de vídeo-tutorials per a professors usuaris de Classpip).

2.2. Arquitectura del projecte

L'arquitectura d'aquest projecte va ser escollida per en Ferran, qui va començar aquesta eina i va escollir les tecnologies que la fan funcionar. Els components més importants d'una aplicació, en general, són:

- Lògica de negoci
 - Part més important de l'aplicació
 - Defineix els processos que involucren a l'aplicació
 - Conjunt d'operacions requerides pel proveir el servei
- Administració de les dades
- Interfície

L'aplicació utilitza diverses tecnologies pel funcionament de les diferents plataformes i la seva interconnexió. Aquestes tecnologies tenen alhora diverses utilitats i ajuden a l'aplicació a connectar les parts que la conformen: Interfície, API i Base de dades.

En la figura 2.1 es mostren les tecnologies esmentades. De cara a programació, les més importants a conèixer són:

- Strongloop: maneig de l'API.
- Angular Material: tecnologia emprada per la interfície del *Dashboard*
- Cordova/ionic: treballa la interfície de l'aplicació pels mòbils Android.
- GitHub: un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font (SCM). El seu funcionament s'explica amb més detall tant al Capítol 3.4. com a l'Apèndix C.

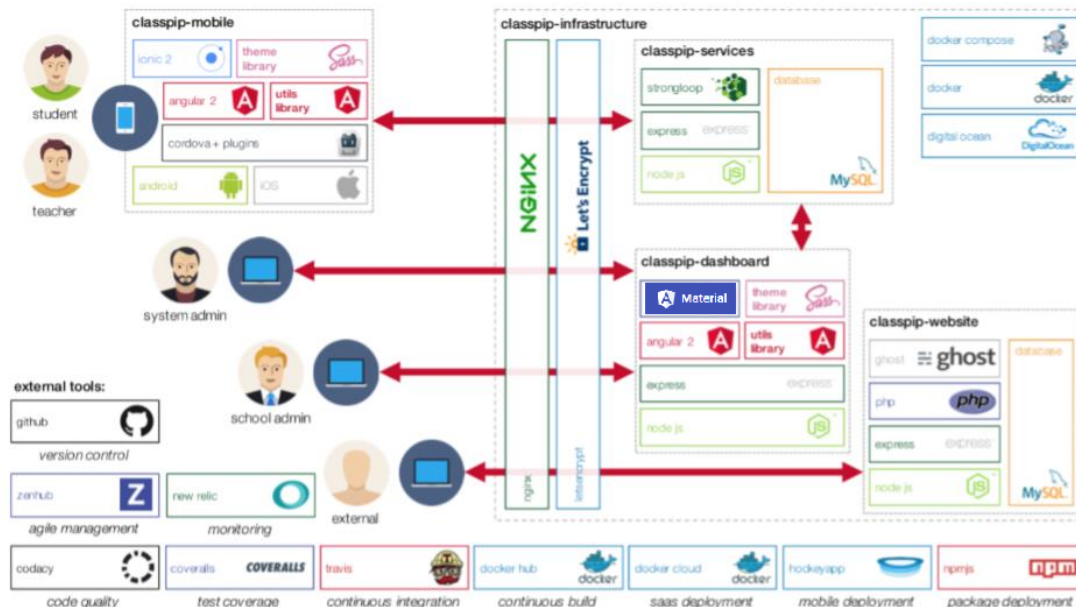


Figura 2.1: Esquema de l'arquitectura de l'aplicació

2.3. Descripció de l'aplicació

Com ja s'ha comentat en anteriors apartats, és un projecte en el que han col·laborat un grup de persones en diferents treballs de fi de grau. L'estructura del projecte va ser creada per en Ferran, qui va escollir l'entorn de treball i el llenguatge de programació del projecte. Posteriorment va comptar amb altres col·laboracions. Aquestes col·laboracions han aportat des de mòduls nous fins a la creació d'una versió vendible. En la figura 2.2, es mostren totes les participacions i les seves aportacions.

Per altra banda cal conèixer què és un repositori per tal d'entendre l'estructura de l'aplicació de Classpip. Un repositori és un conjunt d'arxius amb diverses funcionalitats i, en aquest cas, fins i tot recollits en mòduls, que resulten englobats en cadascuna de les parts fonamentals que permeten a Classpip funcionar correctament. L'adequada connexió entre repositoris permet a Classpip desplegar les funcionalitats implementades en l'entorn de codi amb el qual s'ha treballat. L'estructura de l'aplicació es troba dividida en tres repositoris diferents, cadascun amb la seva funcionalitat particular:

- Services: Ens permet tenir un servidor local, estableix comunicacions amb els mètodes definits als altres repositoris que es troben depositats a l'API.
- Dashboard: Permet veure el contingut de l'aplicació des de l'explorador web.
- Mobile: Permet veure el contingut de l'aplicació com si ho féssim des d'un smartphone.

Tant el repositori de Dashboard com el de Mobile requereixen comunicació amb el de Services per tal de poder utilitzar tots els mètodes definits en els arxius de codi i establir una correcta comunicació amb la base de dades. Només d'aquesta manera hi podrà haver un flux de dades correcte en la interfície d'usuari de l'aplicació.

Les funcionalitats de l'eina estan dividides per mòduls que s'expliquen als propers apartats. Cadascun aporta funcionalitats diferents i combinats entre ells permeten obtenir una eina per a la gamificació. A continuació s'explicarà en què consisteixen els diferents mòduls que conformen l'aplicació de Classpip. S'han ordenat de manera cronològica segons la seva creació.

Proyectos realizados	
	Arquitectura software Ferran Vila
	Puntos e Insignias Guillermo Riverola PBL en una arquitectura de gamificación para un entorno docente
	Colecciones Manuel Mendez Gamificación d'entorns docents
	Preguntas Daniel Viles Plataforma de gamificación docente amb smartphones
	Competiciones Maria del Mar Olmo Generador de competiciones en una herramienta de gamificación
	Primera versión vendible Anna García y Joan Valverde Integració de diferents mòduls i posada a punt d'una eina de gamificació

Figura 2.2: Projectes realitzats prèviament

2.3.1. Grups

Permet al professor gestionar els grups d'alumnes que té assignats. Cada classe sencera, és a dir, cada grup d'una assignatura que tingui al seu càrrec, podrà ser visualitzat i gestionat a la interfície.

2.3.2. Punts i insígnies

Amb aquest mòdul el professor pot premiar els èxits dels alumnes mitjançant punts i insígnies. El primer recompensa els èxits del dia a dia (puntualitat, exercicis, qüestionaris, etc), mentre que el segon premia reptes a llarg termini. Aquest mòdul va ser creat per Guillermo Riverola en la segona versió del projecte. Cal esmentar de manera destacada que en aquesta darrera versió s'ha decidit seguir treballant amb el concepte de punts però, pel que respecta a les insígnies, s'ha decidit deixar-les en un segon pla pel que respecta al seu desenvolupament. Això es deu a que s'ha qüestionat la funcionalitat d'aquestes i no seran utilitzades fins que es trobi un nou sentit per a elles, ja que fins ara únicament representaven tipus de punts determinats amb una imatge (de la mateixa manera que els punts actualment) i generava una ambigüitat de significat.

2.3.3. Col·leccions

Aquest mòdul permet al professor premiar els alumnes amb Cromos que pertanyen a Col·leccions prèviament assignades al Grup de l'alumne. D'aquesta manera, l'estudiant té l'al·licient de realitzar correctament les seves tasques per tal d'aconseguir completar la col·lecció de cromos. Aquest mòdul va ser dissenyat per Manuel Mendez.

2.3.4. Qüestionaris

És un mòdul que permet avaluar un estudiant a partir de la creació de qüestionaris. Va ser el primer mòdul creat per en Daniel Viles amb la idea de gamificar (aplicar mecàniques dels jocs a l'entorn educatiu).

2.3.5. Competicions

Permet al professor crear competicions entre els alumnes. Aquest mòdul va ser desenvolupat per Maria Del Mar sobre el projecte de Ferran, com a projecte independent del de l'Anna i en Joan. Dins del mòdul de Competicions i trobem quatre tipus de competició:

- **Lliga Individual:** Els alumnes participen individualment en una competició amb format Lliga. Segons el resultat de cada jornada, l'alumne aconsegueix punts que es van acumulant a una classificació de la competició. Finalment guanyarà qui hagi obtingut una major quantitat de punts al llarg del curs, i s'establirà un podi entre els tres primers.

- **Lliga Equips:** Els alumnes participen per equips en una competició amb format Lliga. Segons el resultat de cada jornada, l'equip aconsegueix punts que es van acumulant a una classificació de la competició de la mateixa manera que en la individual.
- **Torneig Individual:** Els alumnes participen individualment en una competició amb format Torneig de Tennis. Segons el resultat de cada jornada, el guanyador segueix a la competició, mentre que el perdedor l'ha d'abandonar. Guanyarà qui es mantingui invicte durant tot el torneig. Pel que respecta al format de decisió de cada jornada, hi haurà diverses opcions tal i com s'explica al Capítol 4. Per exemple, una jornada guanyarà l'estudiant que hagi acumulat més punts en una classificació global de Lliga, o es decidirà aleatòriament, o el professor podrà seleccionar el guanyador manualment segons cregui convenient.
- **Torneig Equips:** Els alumnes participen per equips en una competició amb format Torneig de Tennis. Segons el resultat de cada jornada, l'equip guanyador segueix a la competició, mentre que el perdedor ha d'abandonar la competició, tal i com es feia al Torneig Individual.

2.3.6. Equips

Permet al professor crear equips d'alumnes dins d'un grup. D'aquesta manera, aquest mòdul haurà d'estar molt ben relacionat amb el mòdul de grups per tal que el professor pugui crear equips dins dels grups de classe que gestiona, permetent que aquests equips participin més tard en competicions. Aquest mòdul va a ser implementat per Maria Del Mar Olmo.

2.3.7. Assistència

Aquest mòdul es nodreix del mòdul de Punts i Insígnies i permet al professor premiar amb punts els alumnes que arriben puntuals. Aquest mòdul ha estat creat durant aquesta versió del projecte i les seves funcionalitats s'han implementat a aquesta darrera versió. Al Capítol 4.6. s'aprofundeix en la mecànica del seu funcionament.

2.4. Objectius

Els objectius d'aquest projecte han estat dividits en diferents fronts: lliurar un conjunt de materials d'Onboarding per facilitar la feina als futurs programadors, unir tots els mòduls ja existents d'aquesta plataforma amb la finalitat de crear una versió millorada i ensenyar als professors a utilitzar l'aplicació. A continuació es descriuen de forma més detallada els principals objectius.

- Materials per Onboarding:

Els materials d'Onboarding desenvolupats en aquest projecte són un conjunt de manuals i tutorials que pretenen facilitar la feina als futurs programadors.

El material d'aquesta versió en concret inclou la millora de dos manuals ja existents. Per una banda el manual d'instal·lació de la plataforma i de l'entorn de desenvolupament i, per l'altra, el tutorial anomenat *Mesa*, que explica com desenvolupar una funcionalitat en l'aplicació amb l'objectiu d'aprendre a programar en l'entorn de desenvolupament sense tenir nocions de programació prèvies en els llenguatges emprats.

A més a més, s'han desenvolupat tres tutorials bàsics per a programadors principiants, creats amb l'objectiu de proporcionar els coneixements previs necessaris per realitzar el tutorial *Mesa* de forma efectiva.

Per acabar, s'ha preparat una breu explicació de com treballar en equip amb la plataforma del GIT i s'ha realitzat una ampliació del catàleg d'errors amb els problemes que s'han trobat en la realització d'aquest projecte.

- Millora i connexió dels mòduls ja existents i creació de nous mòduls:

La plataforma de Classpip està formada per un conjunt de mòduls independents amb diferents funcionalitats. Un dels objectius d'aquest projecte ha estat connectar tots els mòduls en una mateixa aplicació i afegir les funcionalitats necessàries per presentar una versió vendible d'aquesta. Aquestes funcionalitats sorgeixen juntament amb la millora i connexió dels mòduls ja existents i la creació de nous mòduls.

Totes les noves funcionalitats han estat implementades en el repositori de **Dashboard**, ja que els mòduls de Competicions i Equips no existeixen en el repositori de Mobile. Per tant, s'ha optat per realitzar una feina més extensa i concentrada en un sol repositori, per tal de poder desenvolupar una versió vendible més avançada. La tasca corresponent a adaptar el repositori de Mobile i actualitzar-lo fins el nivell del Dashboard es planteja per a una futura versió, tal i com consta al llistat de futures implementacions realitzat a les Conclusions d'aquesta memòria.

A continuació es pot veure una llista de les millores afegides en aquesta versió de Classpip:

- Unió de les dues versions separades inicials
- Connexió Equips amb la resta de mòduls
- Canvi en el model de Punts
- Classificació per Punts dels alumnes i Equips

- Intercanvi de cromos
- Premis d'assignació manual en Competicions
- Premis automatitzables en Competicions
- En Competicions, opció de decidir guanyador segons els Punts de la classificació general
- Visualització d'Equips
- Mòdul d'Assistència

Aquestes millores es poden trobar explicades detalladament al Capítol [4](#).

- Projectió externa:

S'han preparat un conjunt de vídeo-tutorials per a professors amb l'objectiu d'explicar les diferents maneres d'utilitzar l'aplicació.

Es tracta d'un material d'ajut als professors amb l'objectiu d'aprendre a utilitzar l'aplicació segons el mode al qual es vulgui jugar. Aquests vídeo-tutorials expliquen tres maneres diferents d'utilitzar l'aplicació de Classpip. Una d'elles és jugar únicament amb el mòdul de Punts i Insígnies, una altra és només crear Competicions i la darrera consisteix en crear equips i jugar amb ells. La idea principal és que els professors aprenguin a utilitzar l'aplicació de forma senzilla i entenguin que no és necessari utilitzar tots els mòduls alhora, si no que l'aplicació es pot adaptar a les necessitats de cada classe de forma independent.

Un cop detallada l'estructura d'aquest projecte i de l'aplicació, i anomenats els objectius que s'han buscat en el seu desenvolupament, es passa a mostrar detalladament les millores que aquesta versió ha aportat al projecte de Classpip. En el Capítol [3](#) hi podeu trobar explicades les millores i actualitzacions aplicades al material d'Onboarding i a la documentació que envolta l'entorn de treball del projecte.

CAPÍTOL 3. MATERIALS PER ONBOARDING

Amb l'objectiu de facilitar la feina als futurs programadors d'aquest projecte s'han creat un conjunt de manuals i tutorials anomenats **Materials d'Onboarding**. Les millores aportades a aquest material d'aprenentatge, que han estat un dels principals objectius d'aquest projecte, es troben descrites en aquest capítol. Tot el material d'onboarding amb les millores incloses en aquesta versió es troba a l'**Apèndix**.

3.1. Millores en el manual d'instal·lació de l'entorn de desenvolupament

En el manual d'instal·lació de l'entorn de desenvolupament s'indiquen els passos a seguir durant la instal·lació d'aquest entorn, així com els possibles errors que es poden anar trobant depenent del sistema operatiu de l'ordinador en que es fa la instal·lació.

S'han ampliat totes les directrius i indicacions que ja es donava en la versió anterior del projecte (realitzada per Anna i Joan) el qual es va fer a partir del manual existent del repositori inicial:

<https://github.com/classpip/classpip/wiki/Setup-Dev-Environment>

Al manual de base es va dur a terme la instal·lació en els següents sistemes operatius:

- Windows 10 Pro, versió 1709
- Windows 10 i 8.1
- Ubuntu 17.10

En aquest projecte la instal·lació s'ha realitzat correctament en **Windows 8.1 i Windows 10**, tret d'alguns errors que es tracten en el Catàleg d'Errors de l'Apèndix **D**.

L'entorn de la plataforma no és fàcil d'instal·lar i pot comportar tot un repte depenent de la màquina on s'executi. A més a més, tots els components de la plataforma han de tenir unes versions concretes, ja que molts d'ells depenen de versions determinades d'altres elements i actualitzar només un a la darrera versió pot comportar que la instal·lació falli.

La millora d'aquest manual s'ha dut a terme degut a que, tot i que la versió inicial d'aquest era completament funcional, no s'aturava en explicar detalls i podia provocar dubtes i errors innecessaris per culpa d'explicacions pobres o no adequades per a nous programadors.

L'aportació d'aquest projecte al manual d'instal·lació consisteix en la millora de la redacció i la claredat de les explicacions del procés, un aprofundiment en l'estructura de l'aplicació (divisió en versions de services, dashboard i mobile, amb una breu explicació del funcionament de cadascun) i una ampliació d'aspectes problemàtics en la iniciació a aquest projecte. *Aquest manual es troba a l'apèndix **A** dels annexos.*

3.2. Passeig pel projecte

Uns dels principals problemes per a un nou programador que comença a treballar en el projecte són el desconeixement dels programes, llenguatge i directoris de treball. Per aquest motiu, s'ha proposat realitzar uns tutorials com a suport bàsic per a futurs programadors.

A continuació, s'expliquen tres tutorials diferents, que s'han creat en format de vídeo, per ajudar a l'alumne a situar-se en les diferents tasques que haurà de fer durant el projecte. Són tutorials d'iniciació i, per tant, serveixen a l'alumne per tal d'ajudar-lo a tenir el primer contacte amb el projecte.

3.2.1. Posada en marxa de l'aplicació

En aquest primer tutorial s'explica com posar en marxa l'aplicació la primera vegada per tal de començar a visualitzar-la i jugar amb ella. Es mostren els passos a seguir en les diferents carpetes associades a cada repositori. També es detallen les ordres necessàries i on s'han de executar. Aquest tutorial es troba a la plataforma *streaming Youtube* i es pot accedir amb el següent link:

https://youtu.be/wqSxdP__VQ

3.2.2. Canvi en una pàgina html

L'objectiu d'aquest segon tutorial és ajudar a comprendre els diferents components de les pàgines html. Aquestes pàgines html són sensibles i errors mínims poden provocar errors en tota l'aplicació, fins el punt que l'aplicació no s'obri. Per tant, s'ha de conèixer l'estructura de la pàgina sencera per no cometre errors d'ordre. Cadascun dels components té un inici i un final en el codi, i en cas de no tancar-los en el lloc indicat, l'aplicació no funcionarà correctament.

En aquest tutorial s'explica com realitzar petits canvis de visualització d'una de les pàgines. Es mostra com s'afegeixen diferents elements i com se'ls assignen diverses propietats:

- Executar funcions
- Obrir noves pàgines
- Consultes a l'API

Realitzant aquest tutorial, el programador aprendrà a tractar les pàgines html i les diferents opcions que *Material Angular* ofereix per personalitzar la pàgina. Aquest tutorial es troba a la plataforma *streaming Youtube* i es pot accedir amb el següent link:

https://youtu.be/_-QlUcM7a98

3.2.3. Canvi en una funció

Les funcions, també anomenats mètodes, sustenten l'aplicació i treballen amb les dades relacionant-les amb la interfície. Sense elles, l'aplicació només tindria elements sense accions i contingut que s'obté de la base de dades. L'objectiu d'aquest tutorial és mostrar l'estructura d'una funció, els seus elements (variables, inputs, outputs, etc) i com es relacionen amb l'html. Es tracta d'una primera aproximació senzilla a les funcions, sense entrar en detalls que puguin complicar la comprensió de la seva utilitat.

També s'explica com utilitzar Visual Basic Studio per comprendre millor el codi, llegir les variables i detectar possibles errors. Aquest tutorial es troba a la plataforma *streaming Youtube* i es pot accedir amb el següent link:

<https://youtu.be/LwgDDqn8XpE>

Amb l'ajuda d'aquests tutorials, l'alumne aconsegueix situar-se i entendre la metodologia de treball.

3.3. Milliores tutorial Mesa

Per tal de poder facilitar la iniciació al desenvolupament del projecte, en anteriors versions d'aquesta eina per a la gamificació es va preparar un manual amb les nocions bàsiques en l'entorn de desenvolupament de l'aplicació. En aquest manual s'explicava com crear un nou objecte i com es podia mostrar en una pàgina tant de l'aplicació mòbil (repositori Mobile) com de la versió web (repositori Dashboard). Els passos descrits són els essencials per començar a crear funcionalitats i saber endinsar-se en l'estructura del codi.

D'aquesta manera, s'especifica com comprovar les dades introduïdes a l'aplicació mitjançant la pròpia API de la plataforma o mitjançant el programa Postman, que es basa en fer peticions a la API. Sobretot, el més important que s'intenta transmetre al tutorial original és com fer-ho per poder intercanviar dades entre diferents pàgines, interrelacionar models, retornar-los i guardar i utilitzar dades introduïdes per l'usuari a la interfície de l'aplicació.

El manual permet començar a familiaritzar-se amb el codi del projecte, el llenguatge de programació i la implementació de noves funcionalitats de l'eina. Inicialment tenia una manca d'explicacions necessàries per a programadors principiants, sense les quals el manual deixa de ser efectiu ja que no explica el propòsit de molts dels passos a seguir, generant dubtes i incertesa en no entendre per què s'estan duent a terme aquestes accions en el codi.

En aquesta versió del projecte, s'ha afegit explicacions sobre el funcionament dels models i l'estructura del codi. A més a més s'han clarificat els directoris o versions dels que s'ha de partir per començar a seguir el tutorial (versió d'en Ferran). Per acabar, s'han renovat certes imatges del codi amb mala definició per unes de noves en un format més adient.

En el projecte de l'Anna i el Joan es va entregar aquest manual a dos estudiants per a que el realitzessin, plantejant-los les següents preguntes:

- T'ha semblat útil?

- Trobes alguna cosa a faltar en aquest manual?
- T'has trobat algun error?
- Què es pot fer per millorar-lo?

Els comentaris que es van rebre van ser els següents:

- Calen referències més clares d'on estan guardats els fitxers que s'estan manipulant en cada cas.
- És necessari millorar l'explicació de com introduir imatges en el codi.

En aquesta versió del manual s'ha procurat satisfer aquestes peticions per tal de poder oferir un procés d'aprenentatge més i més útil en cada versió per als nous membres del projecte.

El tutorial complet es troba en l'Apèndix [B](#).

3.4. Treball en equip amb el git

Git és un sistema de control de versions distribuït que permet el desenvolupament de diferents branques d'un mateix projecte de manera simultània, i que després permet realitzar la seva fusió de manera còmoda i útil. A més a més és un entorn de treball multi-plataforma, de manera que es pot fer servir sense restriccions en els sistemes operatius més comuns, com són Windows, Linux o MacOS.

Gràcies a Git es poden anar guardant els canvis que es realitzen en els diversos arxius del projecte, podent anar enrere o endavant en ells després amb unes simples ordres. Aquest conjunt d'ordres bàsiques necessàries per al correcte treball en equip d'aquest projecte es troben explicades a l'Apèndix [C](#).

Per al projecte de Classpip s'ha utilitzat Github, que és un servei d'allotjament de repositoris gestionats amb Git.

3.4.1. Els repositoris de Classpip

En el cas de Classpip, tal i com ja s'ha mencionat en apartats anteriors, es treballa amb tres diferents vessants i per cada una d'aquestes es crea un repositori:

- Dashboard: <https://github.com/alejandromartincruz/classpip-dashboard>
- Mobile: <https://github.com/alejandromartincruz/classpip-mobile>
- Services: <https://github.com/alejandromartincruz/classpip-services>

En cas d'un projecte individual, en cada repositori l'usuari es veu obligat a tenir emmagatzemades les diferents versions treballades anteriorment per si un desenvolupament falla i cal tornar a una versió anterior funcional. En canvi, en cas de fer servir el Git, aquest

permet treballar en funcionalitats noves, mentre hi ha una versió completament operativa i independent. Totes les versions tenen associades diverses branques amb funcionalitats diferents. De manera permanent, al nostre projecte hi podrem trobar definides dues branques imprescindibles:

- **Master:** La versió per defecte i completament operativa, la qual és de producció i on es pujaran en última instància totes les millores funcionals de cara a una versió vendible.
- **Dev:** Una còpia de la branca Master. La seva funció és provar els nous canvis que es volen implementar abans de modificar la versió Master. Si es detecten conflictes, s'evita que l'aplicació quedi inutilitzada.

A aquestes s'afegeixen totes les de les diferents versions. En la figura 3.1 es mostren les branques del projecte.

Per a noves versions, les branques s'han de crear a partir de la branca Master. Una vegada treballada aquesta nova versió, s'ha de fer una primera prova de compatibilitat combinant-la amb la branca Dev. Si el resultat és satisfactori i es vol deixar la versió com a definitiva, es combina amb la branca Master com a últim pas. D'aquesta manera, s'evita que una mala versió afecti a la versió funcional anterior.

Default branch				
master	Updated a month ago by alejandromartincruz		Default	
Your branches				
mejorasLove	Updated 20 days ago by love10fcb	0 24	New pull request	
newramaLove	Updated 28 days ago by love10fcb	0 21	New pull request	
Lovev3	Updated a month ago by love10fcb	5 6	New pull request	
Lovev2	Updated a month ago by love10fcb	5 5	New pull request	
Active branches				
Automations_tournament	Updated 2 hours ago by SergiN96	0 31	New pull request	
home_rebuild	Updated 7 days ago by alejandromartincruz	0 8	New pull request	
Sergi_laBuenav2	Updated 9 days ago by SergiN96	0 30	New pull request	
alejandro	Updated 10 days ago by alejandromartincruz	0 16	New pull request	
dev	Updated 11 days ago by SergiN96	0 42	New pull request	

Figura 3.1: Branques Dashboard Classpip

S'ha creat un manual en col·laboració amb Alejandro Martin Cruz per tal de treballar amb el Git. L'Alejandro va ser qui va implementar la metodologia de treball en equip amb el GIT al projecte. En aquest manual, adjuntat a l'Apèndix C, s'expliquen amb més detall les branques i es llisten totes les ordres que es necessiten per pujar, baixar o renovar versions al GitHub.

3.4.2. Git en equip: metodologia i resultats

En un projecte amb múltiples programadors és molt important la correcta coordinació de tots els participants. Es crea una nova branca per cada millora o funcionalitat extra i es van unint a mesura que avança el projecte. Aquesta unió pot provocar conflictes en el cas que dos persones treballin en el mateix mòdul. Aquests conflictes s'han de resoldre per assegurar el correcte funcionament de l'aplicació. És molt important conèixer i identificar la branca del projecte que té totes les noves millores realitzades en aquest projecte. En cas d'una mala comunicació, es podrien perdre funcionalitats o millores en el procés. Les actualitzacions s'acumulen en cada versió, d'aquesta manera es va assegurant en cada millora la inexistència de qualsevol tipus de conflicte.

Els resultats amb el Git han estat plenament satisfactoris, ja que ha permès treballar en paral·lel en diferents versions o millores. Resulta senzill utilitzar aquesta eina una vegada entesa la mecànica i les principals instruccions utilitzades, que s'expliquen en l'apèndix [C](#).

3.4.3. Posada en marxa de la darrera versió de Classpip

La darrera secció de les actualitzacions del material d'Onboarding, pel que a la utilització del Git respecta, consta d'un petit tutorial que permet la posada en marxa de la darrera versió de Classpip mitjançant el Git. En l'esmentat tutorial es pretén explicar de forma detallada els passos necessaris que cal seguir per tal de poder utilitzar la versió de Classpip que s'ha desenvolupat en aquest treball. D'aquesta manera, si nous membres del projecte volen testejar la interfície o fer-hi proves per tractar d'enfocar les noves millores del seu projecte, hauran de seguir aquest petit tutorial per tal de fer-ho. Així mateix, també servirà a professors que vulguin provar-la i començar-hi a jugar, donat que aquesta versió ja presenta una aplicació vendible amb unes funcionalitats pròpies d'una eina per a la gamificació. Els passos a seguir es poden trobar a l'Apèndix [C.3.](#).

3.5. Ampliació catàleg d'errors

El catàleg d'Errors és un llistat dels errors més comuns trobats durant el procés d'instal·lació i el desenvolupament del projecte de Classpip al llarg de totes les versions des de la creació de l'aplicació. Hi consta una breu descripció de cada error trobat i la seva solució segons el sistema operatiu en el qual s'hagi produït. Es tracta d'un manual dinàmic en el qual si els futurs desenvolupadors troben algun error durant la instal·lació de l'entorn que no està descrit (o es troba una solució més eficient), es pot ampliar o millorar la informació que hi hagi actualment, tal i com s'ha fet en aquesta versió.

En aquest projecte s'han afegit els errors trobats, principalment durant el procés d'instal·lació de l'entorn de treball però també durant el desenvolupament de l'aplicació, al llistat ja existent. Els nous errors trobats i les seves solucions han estat únicament tractats per al sistema operatiu de **Windows 8.1**, ja que ha estat l'emprat en aquesta fase de desenvolupament de l'aplicació. El catàleg d'errors complet es troba en l'Apèndix [D](#).

Per tant, pel que respecta al material d'Onboarding, podem concloure que les noves incorporacions al projecte trobaran menys dificultats a l'hora d'entendre l'estructura, el funcionament d'aquest i els diferents programes que s'hi utilitzen. Tot aquest material de documentació de l'entorn del projecte podrà i haurà de ser millorat amb cada nova versió de Classpip desenvolupada, sent actualitzat de forma dinàmica per tal de preparar de forma eficient l'aprenentatge dels nous estudiants que participin al projecte. Un cop tractat en aquesta memòria un dels objectius principals que es volien dur a terme (la millora dels materials d'Onboarding) es passa a descriure les millores que s'han realitzat en la pròpia aplicació de Classpip, tal i com s'explica en el capítol següent.

CAPÍTOL 4. MILLORA, CREACIÓ I CONNEXIÓ DE MÒDULS

Un dels principals objectius d'aquest projecte ha estat millorar l'aplicació de Classpip, unificant els mòduls existents, afegint noves funcionalitats i creant nous mòduls. Aquest capítol recull totes les millores que s'han dut a terme i explica amb detall el seu funcionament, juntament amb una descripció general de l'aplicació.

Totes les noves funcionalitats han estat implementades en el repositori de dashboard, ja que els mòduls de Competicions i Equips no existeixen en el repositori de mobile. La seva implementació en aquest repositori queda pendent per a una futura versió, tal i com s'explica amb més deteniment al llistat de futures millores d'aquest projecte que es troba en les conclusions.

El concepte de *connexió de mòduls* vol dir que el primer mòdul utilitza el model del segon mòdul per executar funcionalitats. Per exemple: el mòdul de Competicions està connectat amb el mòdul d'Equips i això vol dir que el mòdul de Competicions utilitza el model Equip per crear les competicions per equips.

La finalitat del capítol és explicar com l'aplicació ha passat de ser un conjunt de mòduls separats a una eina per a la gamificació. La següent taula mostra el llistat de noves funcionalitats implementades:

Millores i noves funcionalitats
Unió de dos projectes inicials: Projecte Anna i Joan amb el projecte Maria del Mar
Connexió Equips en tots els altres mòduls
Canvi de model de Punts
Classificació per Punts dels alumnes per Grup
Classificació per Punts dels Equips
Intercanvi de cromos
Premis d'assignació manual en Competicions
Premis automatitzats en Competicions
En Competicions, opció de decidir guanyador segons els Punts de la classificació general
Visualització d'Equips
Mòdul d'Assistència

Taula 4.1: Millores realitzades al projecte

Abans de començar a parlar de les noves funcionalitats implementades en cadascun dels mòduls, és necessari explicar l'estat de la plataforma després de fer la unió entre els dos projectes (4.1.).

4.1. Unió dels dos projectes inicials

Abans de començar cal deixar clar que inicialment hi havia dos projectes per separat: el primer projecte contenia la connexió entre els mòduls de Competicions i Equips i el segon contenia els mòduls de Punts i Insígnies, Col·leccions, Qüestionaris i Grups.

La llista de mòduls i connexions inicials **sense noves funcionalitats** és la següent:

- Grups
- Punts i Insígnies (connectat amb Grups)
- Col·leccions (connectat amb Grups)
- Qüestionaris (connectat amb Punts i Insígnies, Col·leccions i Grups)
- Competicions (connectat amb Grups i Equips)
- Equips (connectat amb Grups)

El primer pas ha estat unir les dues versions inicials del projecte. En finalitzar aquesta nova versió, el projecte ha quedat amb la següent llista de mòduls i connexions (**amb les noves funcionalitats**):

- Grups
- Punts i Insígnies (connectat amb Grups i **Equips**)
- Col·leccions (connectat amb Grups)
- Qüestionaris (connectat amb Punts i Insígnies, Col·leccions i Grups)
- Competicions (connectat amb Grups, Equips, **Punts i Insígnies i Col·leccions**)
- Equips (connectat amb Grups)

Un cop anomenats els mòduls que formen la plataforma i les seves connexions, cal explicar detalladament cadascuna de les noves funcionalitats. A continuació es dedica una secció a cada mòdul per explicar quin ha estat el punt de partida i quines noves funcionalitats s'hi han implementat.

4.2. Mòdul de Punts i Insígnies

El mòdul de Punts i Insígnies permet al professor assignar Punts i Insígnies als alumnes. Inicialment el mòdul de Punts i Insígnies tenia 3 pestanyes en un mat-tab, les quals contenen les següents funcionalitats (Figura 4.1).

Pestanya *Punts*:

- Crear i eliminar Punts i assignar-los a estudiants
- Visualitzar els Punts existents

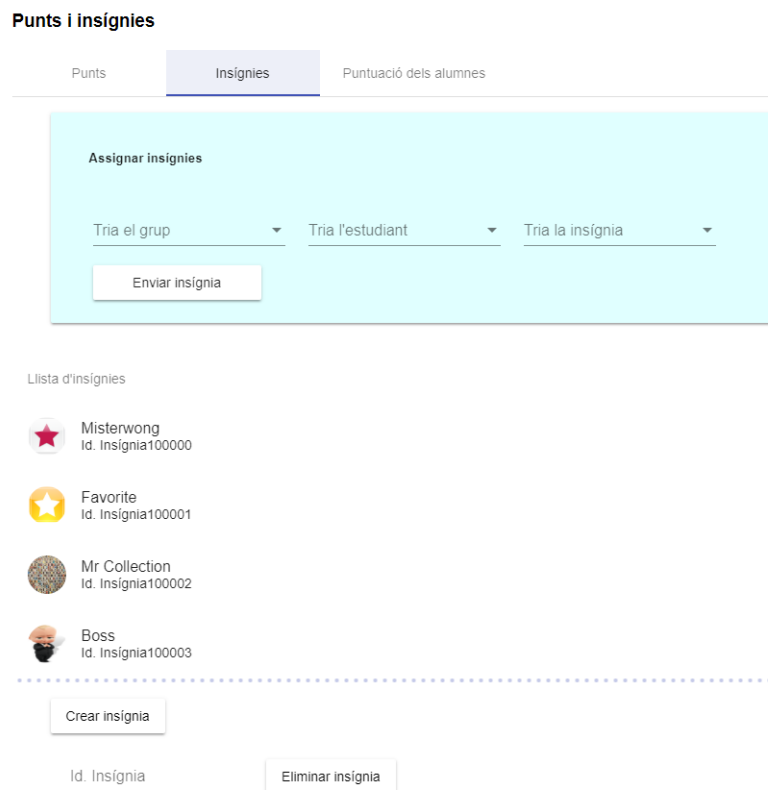


Figura 4.1: Aparença inicial de la interfície del mòdul de Punts i Insígnies

Pestanya *Insígnies*:

- Crear i eliminar Insígnies i assignar-les a estudiants
- Visualitzar les Insígnies existents

Pestanya *Puntuació dels alumnes*:

- Visualitzar una llista dels alumnes d'un grup i el nombre total de punts de cada alumne

A continuació es descriuen les noves funcionalitats del mòdul.

4.2.1. Actualització del model de punts

S'ha millorat el model de Punts per tal de diferenciar-ho amb el d'Insígnies. Per fer-ho, s'ha eliminat la propietat de *image*. Ara els punts tenen les propietats de *name* i *value* (amb el seu respectiu *id*). També s'ha predefinit el valor del punt a *value* = 1.

```
2  "name": "Point",
3  "plural": "points",
4  "base": "PersistedModel",
5  "idInjection": true,
6  "options": {
7    "validateUpsert": true
8  },
9  "properties": {
10   "name": {
11     "type": "string",
12     "required": true
13   },
14   "value": {
15     "type": "number",
16     "required": true
17   }
18 },
19 "validations": [],
20 "relations": {
21   "school": {
```

Figura 4.2: Model Punts i Insígnies: point.json

A part de l'arxiu del repositori de services *point.json*, també s'han hagut d'actualitzar el model i el services de Punts dins de la carpeta de dashboard.

4.2.2. Connexió amb el mòdul d'Equips i millora de la interfície

S'han implementat les connexions amb el mòdul d'Equips de manera que ara es poden assignar Punts i Insígnies als equips.

A continuació es mostra un exemple de com s'assignen premis als equips: en assignar 4 punts de Qüestionari i la insígnia Misterwong a l'equip A, tots els estudiants de l'equip A reben els 4 punts i la insígnia.

El connectar el mòdul d'Equips, s'ha aprofitat per millorar la interfície i fer-la més senzilla i còmode d'utilitzar: ara es té un *mat-card* per sobre del *mat-tab* original, on s'escull l'estudiant o l'equip que es vol visualitzar o assignar-li punts. D'aquesta manera, no s'ha d'escollir tota l'estona l'estudiant o l'equip cada vegada que l'usuari es mogui entre els tabs del *mat-tab*. Es pot veure en la figura 4.3.

Punts i insígnies

The screenshot displays a web interface for assigning points and badges. At the top, there's a section titled 'Punts i insígnies'. Below this, there's a form with the following elements:

- A dropdown menu labeled 'Tria el grup' with 'Group 1' selected.
- A toggle switch labeled 'Individual' (inactive) and 'Teams' (active).
- A dropdown menu labeled 'Tria l'equip' with 'A' selected.

Below the form, there's a horizontal tab bar with three tabs: 'Punts' (selected), 'Insígnies', and 'Puntuació dels alumnes'. The 'Punts' tab is active, showing a section titled 'Assignar punts'. This section contains:

- A dropdown menu labeled 'Tria el punt'.
- A text input field labeled 'Tria la quantitat'.
- An 'Enviar punts' button.

At the bottom of the interface, there's a section titled 'Llista de punts' which shows a table with one row: 'Questionario' and 'Id. Punt: 100001'.

Figura 4.3: Model Punts i Insígnies nova interfície

Per escollir entre l'opció Individual o Equips s'ha posat un Switch ja que només es tenen dues opcions i sempre ha d'estar una d'elles activada.

Els arxius que s'han actualitzat són els que componen la `PointsBadgesPage`: *pointsbadges.html*, *pointsbadges.ts* i *pointsbadges.scss*.

4.2.3. Classificació dels estudiants

S'ha modificat la pestanya de *Puntuació d'alumnes* que fins ara no donava cap informació detallada. Ara mostra una classificació (figura 4.4) per Punts totals dels alumnes d'un grup. També podem veure la classificació dels Equips d'un grup (figura 4.5).

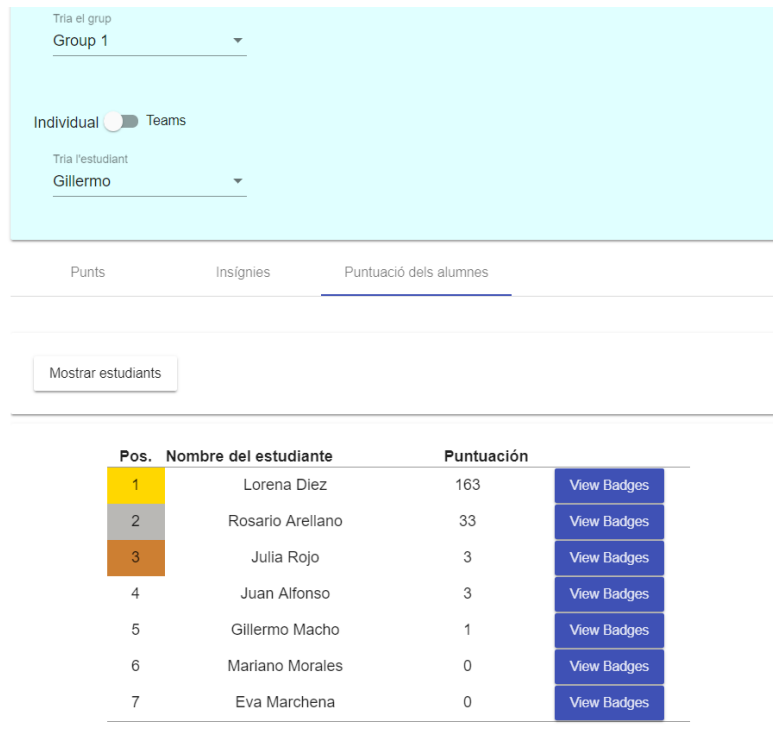


Figura 4.4: Punts de l'alumne

Punts i insignies

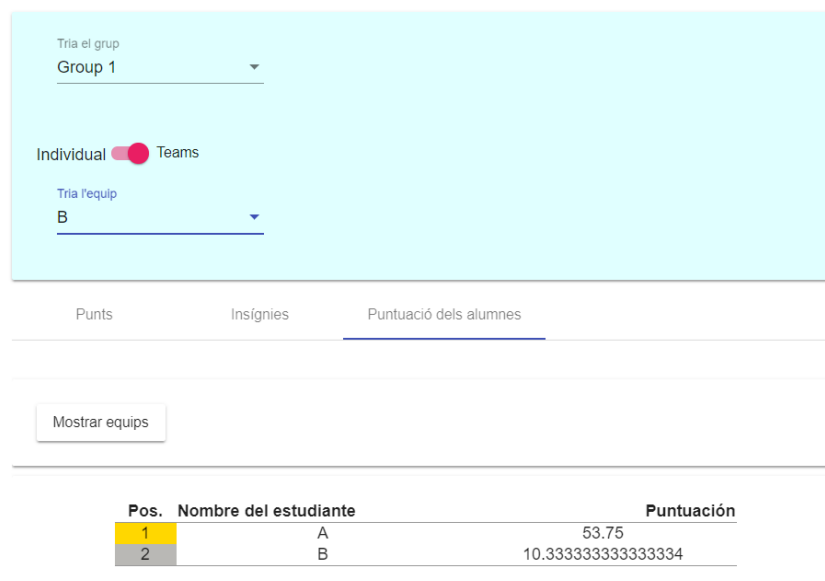


Figura 4.5: Punts de l'alumne

El fer click en el número de punts d'un alumne en la classificació, podem veure (figura 4.6) la descomposició dels Punts totals d'un estudiant i veure els tipus de Punts que se li han assignat, els quals conformen la suma total.

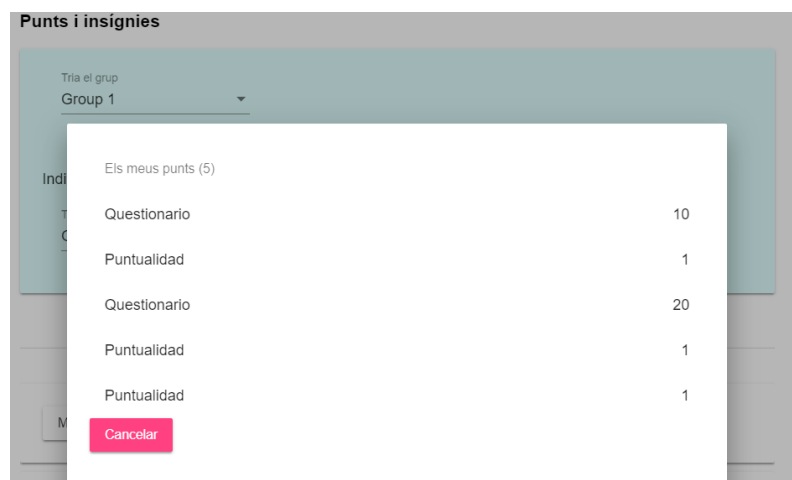


Figura 4.6: Punts dels equips

El mateix passa si cliquem en el botó de *View Badges*, que ens permetrà veure (figura 4.7) les insígnies de l'alumne.

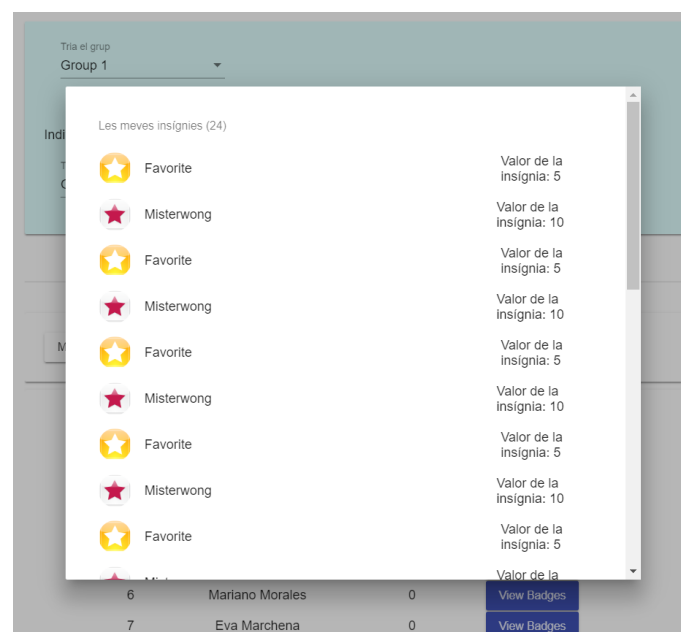


Figura 4.7: Insígnies de l'alumne

La classificació del grup la poden veure tant els alumnes com el professor d'aquest. D'aquesta manera, els professors poden veure i comparar els rendiments dels seus alumnes i els estudiants podran situar-se dins de cada grup en el que es troben.

4.3. Mòdul de Col·leccions

Aquest mòdul permet al professor assignar cromos als alumnes com a recompensa o premi. Els cromos pertanyen a una col·lecció assignada prèviament a un grup d'alumnes d'una assignatura. L'objectiu de l'alumne és completar la col·lecció de cromos.

Abans de les millores realitzades en aquest projecte, el mòdul de col·leccions ja permetia al professor crear noves col·leccions, assignar les ja existents i eliminar-les. Un grup no pot jugar amb cromos si el seu grup no té col·leccions assignades. En la figura 4.8 es mostra la interfície principal de col·leccions, on podem tractar amb les diferents col·leccions que hi ha creades.


Col·leccions

Assignar col·lecció a grup


Tria la col·lecció ▼ Tria el grup ▼

Assignar col·lecció

Llista de col·leccions

 Jugadores FC Barcelona
Id. Col·lecció: 1
Id. Professor: teacher-1

Nombre de cromos de la col·lecció: 20

 Jugadores Real Madrid CF
Id. Col·lecció: 2
Id. Professor: teacher-1

Nombre de cromos de la col·lecció: 4

Crear nova col·lecció

Id. Col·lecció Eliminar col·lecció

Figura 4.8: Pàgina principal de col·leccions

El professor pot modificar cada col·lecció en la seva pàgina: se li permet visualitzar els cromos ja creats, crear-ne de nous o eliminar els ja existents tal i com es mostra a la figura 4.9.

Una vegada una col·lecció és assignada, el professor pot fer quatre assignacions diferents a un alumne específic:

- Assignar un cromo a escollir
- Assignar un cromo aleatori
- Assignar tres cromos aleatoris
- Assignar cinc cromos aleatoris

Des del punt de vista de l'alumne, aquest té accions limitades. Només se li permet veure els cromos que té de cada col·lecció.

Assignar cromos a estudiant

Tria el tipus d'assignació ▼

Tria el grup ▼ Tria l'estudiant ▼

Assignar cromos


Cromos de la col·lecció (20/20)

Ter Stegen

Id. Cromo: 1

Nivell: oro

Ratio d'aparició: bajo



Piqué

Id. Cromo: 2

Nivell: oro

Ratio d'aparició: bajo




Figura 4.9: Pàgina d'una col·lecció específica

Aquest mòdul resultava massa simple per a l'alumne. Per aquest motiu i per motivar la interacció entre els alumnes, s'ha afegit una nova funcionalitat explicada en el Capítol [4.3.1.](#)

4.3.1. Intercanvi de cromos

Aquesta millora permet a dos alumnes negociar amb els cromos i els ofereix la possibilitat de transferir-los entre ells. Des del punt del professor, aquest mòdul no ha sofert cap canvi. Des del punt de vista de l'alumne ara pot al clicar sobre un cromo que ja té i s'obre una nova finestra tal i com es veu a la figura 4.10.

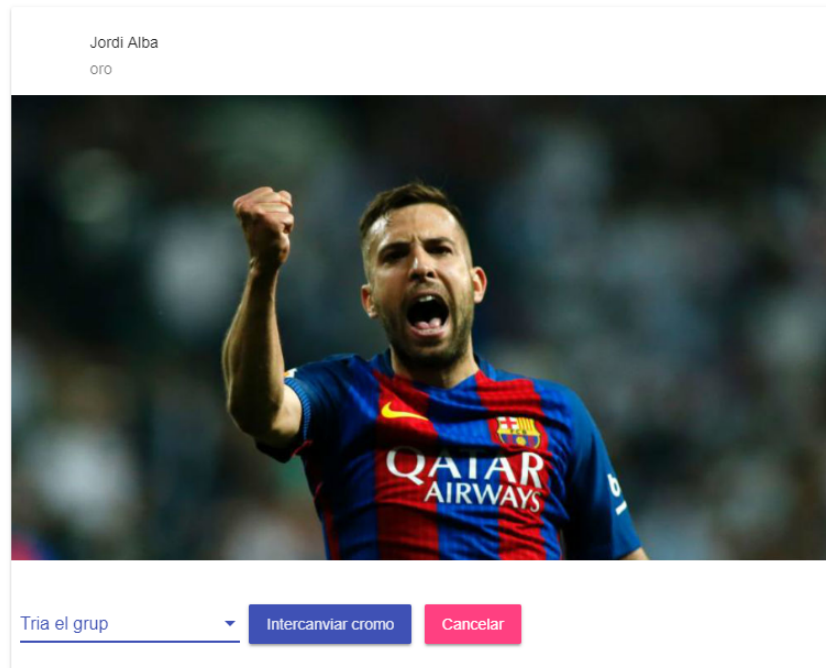


Figura 4.10: Finestra cromo seleccionat

En aquesta finestra, l'alumne pot escollir el company al qual vol enviar el cromo. Només pot seleccionar persones dels grups als qual ell pertany. Una vegada seleccionat el destinatari i realitzat el canvi, el cromo desapareix a l'instant de la col·lecció del remitent i immediatament apareix en la col·lecció del destinatari.

4.4. Mòdul de Competicions

El mòdul de competicions permet al professor crear competicions entre els alumnes, ja sigui del tipus **Lliga** o del tipus **Torneig de tennis**. Els dos tipus de competició estan explicats en el Capítol 2.3.5..

Les funcionalitats ja existents en el moment d'iniciar el projecte eren:

- Pàgina Menú Competició: crear competició, eliminar competició
- Pàgina Crear competició:
 - Tipologia: Lliga o Torneig
 - Tipologia: Individual o per Equips
 - Escollir Jornades i Participants
 - Afegir text informatiu a la competició
- Pàgina Lliga/Torneig:
 - Posar resultat jornades
 - Editar paràmetres de la competició
 - Veure classificació/taula
 - Veure participants i veure jornades/aparellaments.

Les noves funcionalitats estan dirigides a premiar els alumnes que ho fan bé en la competició. D'aquesta manera s'han afegit els **Premis automatitzables** (secció 4.4.2.) i els **Premis assignats pel professor manualment** (secció 4.4.1.). A gran escala aquestes funcionalitats estan basades en la connexió amb els mòduls de Punts i Insígnies i Col·leccions.

Una tercera i nova funcionalitat és la possibilitat de decidir el guanyador d'una jornada a partir dels Punts de la classificació general de cada alumne (secció 4.4.3.).

Abans de començar s'han d'explicar alguns conceptes que tenen en comú les funcionalitats d'assignació de premis.

L'assignació es pot fer per als quatre tipus de competició ja explicats en el capítol 2.3.5.:

- **Lliga Individual:** Els alumnes participen individualment en una competició amb format Lliga.
- **Lliga Equips:** Els alumnes participen per equips en una competició amb format Lliga.
- **Torneig Individual:** Els alumnes participen individualment en una competició amb format Torneig.
- **Torneig Equips:** Els alumnes participen per equips en una competició amb format Torneig.

A continuació s'explica un exemple de com s'assignen premis als equips: el assignar 4 punts de Qüestionari i la insígnia Misterwong a l'equip A, tots els estudiants de l'equip A reben els 4 punts i la insígnia.

En crear una competició, automàticament se li assigna un Punt que tindrà el nom de la competició, i cada cop que s'assignin punts en la competició s'utilitzarà aquest tipus de Punt. És a dir, en crear la Lliga *Liga1* i el Torneig *Torneo2*, es crea el Punt *Liga1* i el Punt *Torneo2* (figura 4.11) . En assignar 3 Punts des de la *Liga1*, els participants premiats rebran 3 vegades el Punt *Liga1* i en assignar 4 Punts des del Torneo2, els participants premiats rebran 4 vegades el Punt *Torneo2*.

Liga1
Id. Punt: 100007

Torneo2
Id. Punt: 100008



Figura 4.11: Punts del tipus Competició

Les següents seccions estan dedicades a explicar les noves funcionalitats d'aquest mòdul.

4.4.1. Premis assignables manualment

Es tracta de l'assignació de Punts, Insígnies i Cromos als participants de la competició de forma manual.

En el cas de competició del tipus Lliga l'assignació manual de premis la pot fer el professor des de la pàgina de Classificació (figura 4.12):

Competiciones > Liga1 > Clasificación

Clasificación de la liga: Liga1						
No.	Nombre del estudiante	Jugados	Ganados	Empatados	Perdidos	Puntuación
1	Lorena Diez	0	0	0	0	0
2	Gillermo Macho	0	0	0	0	0
3	Mariano Morales	0	0	0	0	0
4	Julia Rojo	0	0	0	0	0

Premis:

Tria l'estudiant
Lorena

Punts
Insígnies
Cromos

Número de puntos
Max. 1 xifra

< Volver a la página de la competición

Figura 4.12: Premis tipus Lliga

En el cas de competició del tipus Torneig l'assignació manual de premis es podrà fer des de la pàgina de Tournament (figura 4.13):

Per fer l'assignació primer escollim el participant des d'un mat-select, després escollim el tipus de premi des del mat-tab i finalment escollim els paràmetres del premi: si és un Punt hem d'escollir quantes vegades l'enviem, si és una Insígnia hem d'escollir el seu valor i si són Cromos ho podem fer igual que en la pàgina de Col·leccions.

Hi ha 4 opcions d'assignació de Cromos (les mateixes que en la pròpia pàgina de Col·leccions):

- Assignar un cromos a escollir
- Assignar un cromos aleatori
- Assignar tres cromos aleatoris
- Assignar cinc cromos aleatoris

En el cas d'escollir la primera opció, apareixerà una casella de selecció per escollir el cromos (figura 4.14):

Participantes eliminados:

Eliminados

No hay jugadores eliminados.

Premis:

Equipo

A

Punts

Insignies

Cromos

Valor insignia primer clasi...

Max. 2 xifres

[← Volver a la página de la competición](#)

Figura 4.13: Premis tipus Torneig

Triar l'estudiant

Lorena

Punts

Insignies

Cromos

Triar la col·lecció

Jugadores FC Barcelona

Triar el tipus d'assignació

Assignar un cromó a e...

Triar el cromó

Piqué (oro)

[Assignar cromos](#)

[← Volver a la página de la competición](#)

Figura 4.14: Premis tipus Cromo

4.4.2. Premis automatitzables

Es tracta de l'assignació de Punts, Insígnies i Cromos als participants de la competició de forma automatitzada. El professor escull les opcions d'automatització que vol en el moment de crear la competició i després aquestes funcionen sense haver de fer res més.

En els dos tipus de competició (Lliga 4.15 i Torneig 4.16) podem assignar premis de forma automatitzada de dues maneres: Punts i Col·leccions. El que s'ha buscat és que el professor pugui escollir entre jugar només a Punts, jugar només a Col·leccions, jugar a ambdues o no jugar a res.

L'assignació de Cromos la fa l'aplicació escollint una Col·lecció i un cromó aleatoris.

Creando competición de tipo Liga:

✓ Inicialización — ✓ Participantes — 3 Premis Automatizables — 4 Jornadas — 5 Añadir información — 6 Terminado

Tria els premis

Assignar 1 punt de competició al primer i segon classificats (al finalitzar cada jornada) ☐

Assignar 1 cromó aleatori al primer i segon classificats (al finalitzar cada jornada) ☒

Siguiente

Figura 4.15: Menú de creació de Lliga

Creando competición de tipo Tenis:

✓ Inicial... — ✓ Partic... — 3 Premis Auto... — 4 Jorna... — 5 Añadir in... — 6 Termini...

Tria els premis

Assignar 1 punt de competició als finalistes i un altre al guanyador ☐

Assignar 1 cromó aleatori als finalistes i un altre al guanyador ☐

Siguiente

Figura 4.16: Menú de creació de Torneig

Els premis en el tipus Lliga s'assignen en finalitzar cada Jornada al primer i segon classificats de la competició. En el tipus Torneig s'assignen als finalistes i al guanyador de la final.

En els dos tipus de competició els premis s'assignen en una pàgina en la qual, un cop feta l'assignació, es mostra la classificació/taula actualitzada de la competició (figures 4.17 i 4.18).

En el cas que s'hagin escollit premis d'assignació de Cromos i el Grup associat a la competició no tingui cap Col·lecció assignada, l'aplicació avisa a l'usuari de que s'ha d'assignar una col·lecció primer i no assigna els premis (figures 4.19 i 4.20).

Competiciones > Liga1 > Premis Automàtics

Clasificación de la liga: Liga1						
No.	Nombre del estudiante	Jugados	Ganados	Empatados	Perdidos	Puntuación
1	Lorena Diez	1	1	0	0	3
2	Mariano Morales	1	0	1	0	1
3	Julia Rojo	1	0	1	0	1
4	Gillermo Macho	1	0	0	1	0

S'han assignat els premis automàtics.

< Volver a la página de la competición

Figura 4.17: Pàgina d'automatitzacions Lliga

Competiciones > Copa2 > Seguimiento del torneo

Seguimiento del torneo: Copa2

Este torneo ha finalizado.

Ganador: B

Final:

Finalistas

 A

 B

S'han assignat els premis automàtics.

Participantes eliminados:

Eliminados

Figura 4.18: Pàgina d'automatitzacions Torneig

Competiciones > Liga5 > Premis Automàtics

Clasificación de la liga: Liga5						
No.	Nombre del estudiante	Jugados	Ganados	Empatados	Perdidos	Puntuación
1	Mariano Morales	1	1	0	0	3
2	Eva Marchena	1	1	0	0	3
3	Gillermo Macho	0	0	0	0	0
4	Julia Rojo	1	0	0	1	0
5	Juan Alfonso	1	0	0	1	0

*Los participantes con un número menos de enfrentamientos realizados se debe a la irregularidad de ser un número impar de participantes totales en esta liga. Los descansos que deben realizarse en cada jornada por parte de algún participante conlleva a no sumar puntos en esa jornada.

No s'han assignat els premis automàtics:

No hi ha cap col·lecció de cromos assignada a aquest grup

[← Volver a la página de la competición](#)

Figura 4.19: Pàgina d'automatitzacions Lliga

Competiciones > Copa2 > Seguimiento del torneo

Seguimiento del torneo: Copa2

Final:

Finalistas
 A
 B

No s'han assignat els premis automàtics:

No hi ha cap col·lecció de cromos assignada a aquest grup. No s'han assignat els premis automatitzables.

Participantes eliminados:

Eliminados
No hay jugadores eliminados.

[← Volver a la página de la competición](#)

Figura 4.20: Pàgina d'automatitzacions Torneig

4.4.3. Resultats Jornada a partir del nombre de Punts del participant

En la versió anterior, les possibilitats de decisió dels resultats de les jornades eren limitades. Es podia decidir el resultats de dues maneres diferents:

- Manualment: el resultat és assignat manualment pel professor segons els seus propis criteris. Per exemple, segons el resultat d'un examen fet a classe i comparant els resultats dels dos contrincants.
- Aleatòriament: el resultat s'obté d'una variable *Random* que proposa la pròpia aplicació. No segueix cap altre criteri objectiu.

En aquesta versió, tal i com es veu a la figura 4.21, s'inclou una nova opció que permet al professor decidir el resultat automàticament tenint en compte els progressos de l'alumne a la seva classe. Aquesta nova opció compara els punts generals de cada alumne en total a la seva classe i decideix com a guanyador l'estudiant que té més punts.

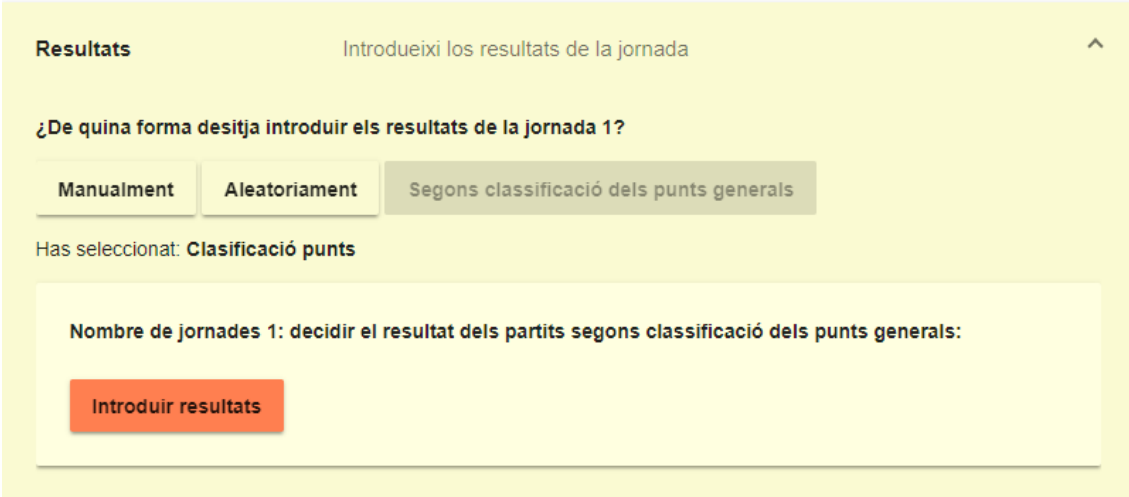


Figura 4.21: Opció de decidir la jornada pels punts en la classificació general

En el cas de lligues, hi poden haver empats en el cas que ambdós estudiants tinguin la mateixa quantitat de punts. En el cas de tornejos, no existeix el cas d'empat. Per tant, en cas d'empat, es fa un sorteig entre els dos estudiants per decidir el guanyador del partit.

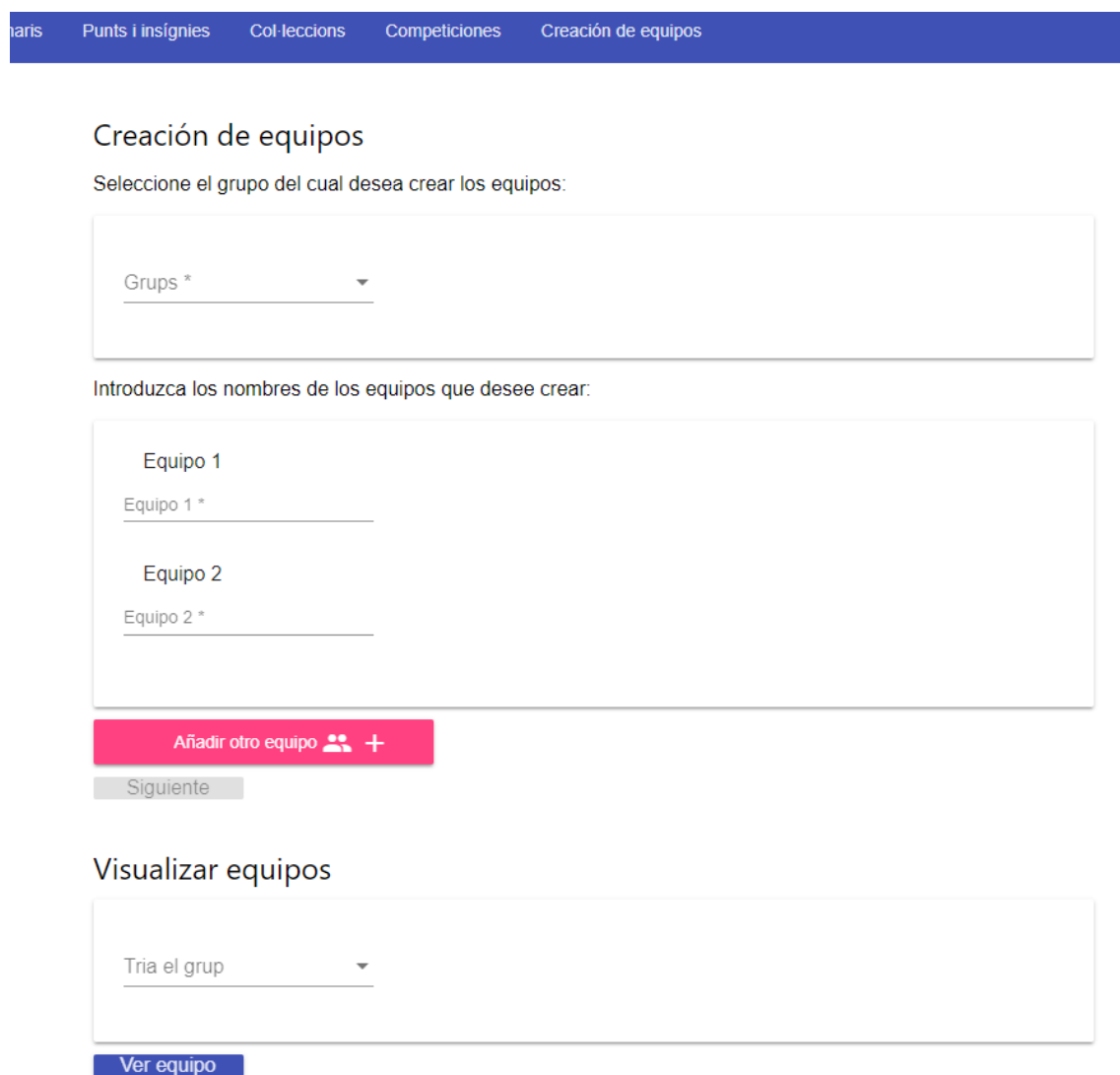
4.5. Mòdul d'Equips

Inicialment el mòdul d'Equips permetia crear Equips i assignar-los a Grups, però la única forma que teníem de visualitzar els estudiants que formen un equip i els equips que pertanyen a un grup era creant una Competició per equips. Mitjançant un botó que apareix dins de la pàgina d'una competició, podem veure els equips que participen en la competició.

Per solucionar aquest problema s'ha afegit la funcionalitat de Visualització d'equips, explicada a continuació.

4.5.1. Visualització d'equips

Aquesta funcionalitat permet al professor visualitzar els Equips existents i els estudiants que els conformen. En la pàgina d'Equips, a sota de la funcionalitat de Creació d'Equips hi trobem la funcionalitat de Visualització d'Equips (figura 4.22).



Inicio Puntos e insignias Colecciones Competiciones Creación de equipos

Creación de equipos

Seleccione el grupo del cual desea crear los equipos:

Grupos *

Introduzca los nombres de los equipos que desee crear:

Equipo 1
Equipo 1 *

Equipo 2
Equipo 2 *

Añadir otro equipo +

Siguiendo

Visualizar equipos

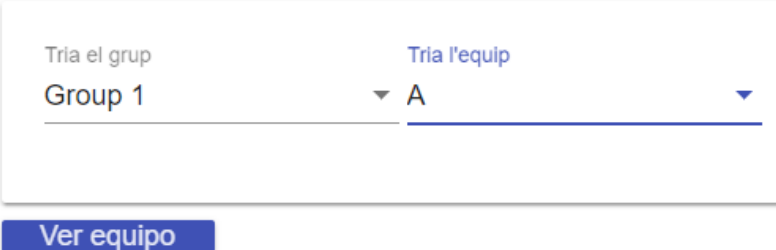
Tira el grup

Ver equipo

Figura 4.22: Menú del mòdul d'Equips

Per visualitzar un equip hem d'escollir el grup, després un dels equips que pertanyen al grup i clicar el botó de Veure Equip (4.23), el qual ens portara a la pàgina on es veuen els estudiants de l'equip (4.24).

Visualizar equipos



The form consists of two dropdown menus and a button. The first dropdown is labeled 'Tria el grup' and has 'Group 1' selected. The second dropdown is labeled 'Tria l'equip' and has 'A' selected. Below the dropdowns is a blue button labeled 'Ver equipo'.

Figura 4.23: Visualitzar Equips

Grupo 1 , Equipo A

Student
Rosario Arellano
Lorena Diez
Julia Rojo
Eva Marchena

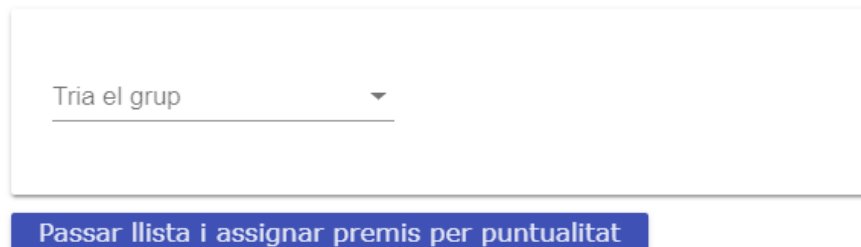
[Volver a creación de equipos](#)

Figura 4.24: Pàgina de visualització de l'equip

4.6. Mòdul d'Assistència

El mòdul d'Assistència (figura 4.26) és un mòdul totalment nou que es nodreix de les funcionalitats del mòdul de Punts i Insígnies i permet al professor premiar als alumnes que arriben puntuals, amb punts del tipus Puntualitat.

Assistencia



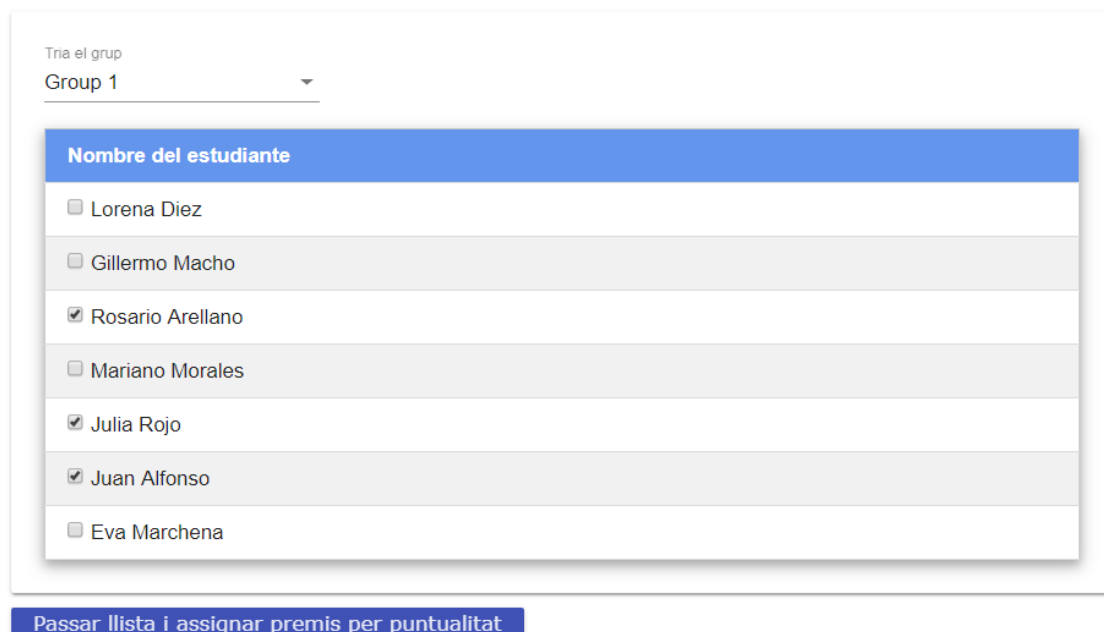
Tria el grup ▼

Passar llista i assignar premis per puntualitat

Figura 4.25: Menú del mòdul d'Assistència

El seu funcionament és molt senzill, primer escollim el grup del qual volem mirar la puntualitat, després seleccionem els alumnes que han arribat puntuals i finalment premem el botó.

Assistencia



Tria el grup
Group 1 ▼

Nombre del estudiante
<input type="checkbox"/> Lorena Diez
<input type="checkbox"/> Guillermo Macho
<input checked="" type="checkbox"/> Rosario Arellano
<input type="checkbox"/> Mariano Morales
<input checked="" type="checkbox"/> Julia Rojo
<input checked="" type="checkbox"/> Juan Alfonso
<input type="checkbox"/> Eva Marchena

Passar llista i assignar premis per puntualitat

Figura 4.26: Menú del mòdul d'Assistència

En prémer el botó els alumnes seleccionats rebran 1 punt del tipus Puntualitat.

Una vegada implementades totes aquestes millores, s'obté el producte final: una aplicació totalment funcional i llesta per ser utilitzada. Per aquest motiu, en el següent capítol es prepara material per facilitar la seva utilització per part del professor.

CAPÍTOL 5. PROJECCIÓ EXTERNA

Un dels nostres objectius han estat millorar les eines de projecció externa ja existents. Per aquest motiu s'han realitzat tres tutorials en format vídeo-tutorial. Aquests tutorials estan enfocats als professors que volen utilitzar aquesta aplicació com a eina a les seves aules. Es tracta de tutorials bàsics que ajuden al professor a familiaritzar-se amb l'aplicació i els seus diferents mòduls, depenent del mode amb el qual es vulgui jugar en cada cas. Per tant, cada professor podrà conèixer com utilitzar de manera independent cada mòdul segons les necessitats de la seva classe. A continuació s'expliquen els diferents tutorials.

5.1. Com jugar amb assignació de punts

En aquest primer tutorial es mostren els passos que ha de seguir un professor per començar a jugar amb els punts. S'explica com es creen nous punts, com s'assignen i com s'eliminen. També s'esmenta com utilitzar la classificació amb qüestions relacionades amb els punts. En aquesta classificació, el professor pot veure d'on provenen els punts que posseeix cada estudiant i, per tant, com els ha obtingut. A més a més s'expliquen diferents tipus punts predefinits, com per exemple els d'Assistència i de Competició, i la seva procedència. Aquest tutorial es troba a la plataforma *streaming Youtube* i es pot accedir amb el següent link:

<https://youtu.be/5BbDPXvUhoo>

5.2. Com jugar a una competició

En aquest segon tutorial es mostren els diversos tipus de competició entre els que pot escollir el professor. D'aquesta manera, s'ensenya un exemple de com crear una competició individual del tipus Lliga. Per altra banda, s'expliquen amb detall els diferents passos que es troben en la creació d'una competició i els diversos tipus de premis automatitzats que s'han afegit en aquesta versió del projecte. Una vegada creada la competició, es parla de les pàgines que té el mòdul de competició. Finalment, s'explica com es juga una jornada en l'aplicació i es mostren les opcions disponibles per tal de decidir-ne el resultat. El tutorial acaba completant totes les jornades de la competició i obtenint el pòdium que resulta d'aquesta. Aquest tutorial es troba a la plataforma *streaming Youtube* i es pot accedir amb el següent link:

<https://youtu.be/X2h8S8tas9U>

5.3. Com crear equips i jugar amb ells

Classpip permet jugar tant individualment com en equips a dins d'un grup. L'objectiu d'aquest tutorial és ajudar al professor a començar a jugar amb equips. Primerament, es mostra com crear equips dins d'un grup d'entre els quals el professor tingui accés. Una vegada creats, es s'explica com visualitzar els components que conformen cadascun dels

equips. Per altre banda es realitza una explicació detallada de com aquests equips poden jugar en altres mòduls, com per exemple en el de Punts i Insígnies. També s'exemplifica com assignar els punts a un equip específic i com visualitzar la classificació d'equips. Finalment ensenya al professor com crear una competició d'equips del tipus Torneig. Aquest tutorial es troba a la plataforma *streaming* **Youtube** i es pot accedir amb el següent link:

<https://youtu.be/RKCT3eixKCo>

Un cop acabat d'esmentar aquest darrer objectiu del projecte es passa a desenvolupar les conclusions d'aquest, tant a nivell tècnic com a nivell personal, les quals podreu trobar al proper i darrer capítol de la memòria.

CONCLUSIONS

Les conclusions han estat redactades a nivell tècnic i a nivell personal. Les conclusions tècniques defineixen els resultats obtinguts a partir dels objectius plantejats en iniciar el projecte i les conclusions personals reflecteixen tant els resultats obtinguts a nivell personal com les impressions sobre el projecte i l'entorn de treball utilitzat.

Les conclusions tècniques es divideixen en funció d'els resultats obtinguts relacionats amb la plataforma de Classpip, el material d'Onboarding i la projecció externa, tal i com s'explica en l'apartat següent.

Plataforma de Classpip

El principal objectiu ha estat presentar una versió millorada de la plataforma Classpip. El punt de partida del projecte han estat dos versions independents (les versions de l'Anna i el Joan i la de la Maria del Mar) desenvolupades sobre la mateixa base, el projecte del Ferran. Per una banda la versió de l'Anna i el Joan implementa els mòduls de Punts i Insígnies i Col·leccions i per l'altre, la versió de la Maria del Mar implementa els mòduls de Competicions i Equips. La unió d'aquestes dues versions ha estat el primer pas per tal d'obtenir una primera versió de la plataforma de Classpip, amb totes les funcionalitats dels mòduls ja existents.

A partir d'aquest punt s'han afegit noves funcionalitats a la plataforma amb l'objectiu de convertir-la en una eina per a la gamificació. La idea principal ha estat aconseguir motivar els alumnes i augmentar la competitivitat entre ells.

La primera millora a gran escala s'ha vist reflexada en afegir les funcionalitats dels mòduls de Punts i Insígnies i Col·leccions al mòdul de Competicions i en el moment d'implementar una Classificació general d'alumnes en funció dels Punts obtinguts, en el mòdul de Punts i Insígnies. Amb aquestes millores l'aplicació fomenta la competitivitat gracies a l'obtenció de premis en forma de Punts, Insígnies o Col·leccions en les Competicions i la visualització d'aquests premis en una Classificació general. Les competicions ofereixen la possibilitat d'assignar premis automatitzables que permeten al professor reduir la seva càrrega de treball durant l'ús de l'aplicació.

Tenint en compte que el treball en equip es considera important en la Gamificació, s'ha connectat el mòdul d'equips amb la resta. D'aquesta manera el professor pot fer competir els alumnes o donar-los punts individualment o per equips, segons la necessitat o la manera escollida de jugar amb la plataforma.

La Classificació general també implementa les funcionalitats del mòdul d'Equips amb la finalitat de que l'alumne pugui visualitzar el seu progrés individual o en equip.

També s'ha pensat en l'ús de l'aplicació a fora de l'aula, amb la implementació de la funcionalitat d'Intercanvi de Cromos. Aquesta funcionalitat permet als alumnes intercanviar Cromos sense la necessitat d'una aprovació per part del Professor. Així s'aconsegueix que els alumnes interactuïn entre ells.

Per acabar, la incorporació del mòdul d'Assistència sorgeix de la idea de premiar els alumnes que arriben puntuals a classe, amb l'objectiu de fomentar la puntualitat.

Material d'Onboarding

El segon objectiu ha estat millorar el material d'Onboarding, el qual és un conjunt de tutorials i manuals que tenen com a objectiu facilitar la feina als programadors de les posteriors versions d'aquest projecte.

Les millores aportades a aquests manuals han estat producte de la experiència obtinguda durant la utilització dels materials per l'Onboarding desenvolupats en projectes anteriors. A partir de les dificultats per participar en el desenvolupament de la plataforma de Classpip s'han afegit millores en els manuals, les quals han estat considerades com a necessàries. A més a més d'aclarir els dubtes sorgits durant l'execució dels manuals, també s'han redactat definicions que faciliten la comprensió i l'enteniment del sentit d'alguns passos a seguir. Per acabar, s'han afegit les correccions necessàries per tal de resoldre els dubtes que sorgien de les enquestes realitzades per l'Anna i el Joan, en les que es demanava feedback un cop realitzats els manuals i tutorials d'Onboarding.

Pel que fa al Catàleg d'Errors, s'han afegit els errors obtinguts durant la instal·lació de l'entorn de treball de la plataforma Classpip.

El desenvolupament de tres vídeo-tutorials que ajuden al futur programador a comprendre les diferents parts del llenguatge de programació també forma part de la nova versió del material d'Onboarding. La realització d'aquests tutorials ha de ser prèvia a la del tutorial Mesa, el qual es considera un tutorial de nivell més avançat.

Finalment s'ha desenvolupat un manual que explica els passos a seguir per poder treballar en grup utilitzant la eina GitHub. La tecnologia del Git permet el desenvolupament d'un mateix projecte en diferents versions de forma paral·lela i és la primera vegada que s'utilitza per a la realització d'una nova versió de la plataforma de Classpip. La utilització d'aquesta eina ha estat gairebé obligatòria degut al nombre de participants d'aquest projecte i la necessitat de treballar en diferents funcionalitats de forma simultània i paral·lela.

Projecció externa

Un altre objectiu important ha estat ajudar als professors a familiaritzar-se amb l'aplicació. En aquest cas s'ha optat per en crear diferents vídeo-tutorials en els que s'explica com utilitzar l'aplicació de diferents maneres. Els tutorials ajuden el professor a entendre els mòduls i a jugar de manera independent a diversos mòduls de l'aplicació de Classpip, com per exemple a Punts o a Competicions, segons les necessitats que cada professor cregui convenients per a la seva classe.

Propostes de futures millores

De cara al futur, aquest projecte presenta àmplies possibilitats. Es proposen les següents millores:

- Un dels més importants objectius per a futures versions de Classpip ha de ser actualitzar el repositori de **Mobile** del projecte al nivell del Dashboard, implementant-hi el mòdul de Competicions i els nous que s'han afegit en aquesta versió de l'aplicació, juntament amb les noves funcionalitats dissenyades.
- Trobar un nou sentit per al concepte d'*insígnies*. En aquesta versió s'ha considerat innecessari seguir implementant millores amb l'actual model d'insígnia, ja que el seu sentit és el mateix que els dels punts (els dos tenen un tipus determinat i se'ls associa una imatge per representar-los). Per tant, seria necessari redefinir el concepte d'insígnia si es vol seguir treballant amb aquesta part del mòdul de Punts i Insígnies i afegir-ne més funcionalitats a l'aplicació.
- Assignar recompenses a llarg termini de manera que premiïn els alumnes segons com hagi estat de satisfactori el seu progrés al llarg de tot el curs en diferents àmbits (puntualitat, puntuació, qüestionaris...).
- Afegir propietat *Data* en els models de *PointRelation* i *BadgeRelation*. Aquesta propietat defineix l'hora exacta en que ha estat assignat un Punt o una Insígnia.
- Afegir en el mòdul de Col·leccions un llistat de cromos repetits. Aquest llistat acumula els cromos que l'alumne rep però ja te.
- Afegir la opció de resoldre una Jornada de Competició a partir de la nota resultant d'un Qüestionari.
- Poder ordenar per data o filtrar per tipologia el llistat de Punts d'un alumne. Aquesta millora està relacionada amb la propietat *Data* esmentada prèviament.
- Afegir la possibilitat d'assignar Cromos a Equips des del mòdul de Col·leccions (el professor ha de poder assignar col·leccions a equips en particular).
- Solucionar el problema de compilació del mòdul de Qüestionaris (dependència circular pel fet de passar dades entre dos qüestionaris diferents).
- Afegir noves automatitzacions en Competicions.

Conclusions personals

A nivell personal, ens hem enfrontat a aquest projecte com un repte, ja que desconeixíem totes les tecnologies emprades en ell, tot i que teníem nocions d'altres llenguatges similars obtingudes al llarg del Grau d'Aeronavegació. S'ha seguit un procés d'aprenentatge durant el projecte, ja sigui seguint estructures fetes per antics programadors com informació obtinguda a Internet i realitzant proves. Cal esmentar que l'anomenada '*corva de productivitat*' no ha començat a ser pronunciada fins als darrers dos mesos. És a dir, que al començament del projecte s'és poc productiu ja que l'aprenentatge és lent i costós, però un cop et trobes familiaritzat amb el codi i l'entorn de treball de l'aplicació cada cop pots desenvolupar millores i actualitzacions de manera més ràpida i efectiva.

El repte va començar aviat amb l'instal·lació de les eines, però els manuals i els catàlegs d'errors van facilitar aquesta tasca. Ens vam trobar la manca de definicions sobre les diferents eines i vam haver de fer una investigació pròpia. En cas de trobar-nos errors d'instal·lació fora del catàleg, va fer falta una recerca pels diferents fòrums disponibles a Internet.

Una vegada vam començar a treballar amb el codi ens vam trobar codis fets per diferents persones i diverses estructures maneres de fer les coses. Vam trobar dificultats a l'hora d'entendre les decisions que s'havien pres fins el moment d'agafar nosaltres el projecte, però tot i així s'ha pogut treballar sobre els codis anteriors correctament i en cas de dubtes majors, vam obtenir el suport dels tutors que ens van proporcionar la informació adient.

Un cop conclòs el procés d'aprenentatge, s'ha tractat de deixar els manuals com a nosaltres ens hauria agradat tenir-los en el moment de començar, amb més detalls i la informació sobre les eines que s'han utilitzat.

En ser un projecte col·laboratiu, s'han trobat moltes decisions ja preses. Com a opinió personal nostra, si es donés l'oportunitat de canviar alguna cosa en l'estructura de l'aplicació, potser hauríem escollit un altre llenguatge de programació amb el qual tinguéssim més experiència els estudiants d'aquesta carrera (com ara el llenguatge C). A més a més, faríem el codi de l'aplicació (sobretot l'html) de forma més robusta i clara però també flexible per tal que admetés canvis amb el pas de les versions que es van desenvolupant al projecte (vam trobar el codi amb una sèrie de petits detalls com ara organització que dificultaven el desenvolupament en ells). Tot i així, trobem que l'estructura del codi i l'entorn de treball ha estat un encert, ja que com hem pogut comprovar un cop familiaritzats amb tot, ens ha permès crear una eina per a la gamificació de manera funcional, que era l'objectiu primordial des que es va començar aquest projecte a gran escala.

El repte més desafiant

Sens dubte el repte més desafiant que ens hem trobat durant el procés de desenvolupament del projecte, ha estat el moment d'aprenentatge autodidàctic dels llenguatges de programació emprats en els repositoris de la plataforma de Classpip.

Abans de començar a implementar noves funcionalitats en la plataforma, has de ser capaç d'entendre el llenguatge de programació i també has d'estar familiaritzat amb la organització i l'estil de programació dels repositoris. La dificultat per assolir aquest nivell de

coneixements és elevada degut a la quantitat de persones que han participat en el desenvolupament de la plataforma de Classpip. A més a més, en el nostre cas no havíem programat mai en aquests llenguatges.

L'aprenentatge autodidàctic s'ha basat fonamentalment en l'estudi dels codis dels repositoris. Estudiant la relació entre els arxius dels repositoris i fent proves hem aconseguit bona part dels coneixements. Per altra banda hem estudiat tutorials que es poden trobar per Internet.

També és necessari comentar que qualsevol petita millora implementada de manera funcional en l'aplicació requereix d'un esforç molt més elevat del que sembla, ja que s'ha d'implementar en diferents arxius de codi de manera simultània per tal que funcioni, amb l'adequada comunicació amb la base de dades. A aquest fet s'afegeix que certes parts del codi de l'aplicació no eren gens robusts i acceptaven millores molt concretes i determinades (com per exemple els fitxers html del mòdul de Punts creats per l'Anna i en Joan), cosa que ha dificultat el desenvolupament de codi en ells i ens ha suposat un repte encara més gran. Per altra banda, aquest desafiament ens ha motivat a l'hora de desenvolupar els materials d'Onboarding, els quals creiem que facilitaran molt la feina als futurs programadors.

En tot cas, podem concloure que la realització d'aquest projecte ha estat un propòsit estimulador que ha posat a prova les nostres capacitats, tant de programació de codi en un nou llenguatge com de treball en equip i organització de versions. Considerem que en ha resultat veritablement útil pel fet d'haver pogut aprendre un nou llenguatge de programació i també per haver pogut experimentar el que és participar en un projecte de desenvolupament a gran escala i amb diferents fronts oberts al mateix temps, el qual requereix una gran coordinació i feina en equip (capacitat imprescindible per a tot aquell que vulgui convertir-se en enginyer).

BIBLIOGRAFIA

- [1] Universitat Politècnica de Catalunya [online]: <http://www.upc.edu/ca>
- [2] Gamificació [online]: <https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/70991.html>
- [3] Strongloop [online] : <https://strongloop.com/>
- [4] Ionic 2 [online]: <http://ionic.io/2>
- [5] Apache Cordova [online]: <https://cordova.apache.org/>
- [6] Web del projecte [online]: <https://classpip-onboarding.herokuapp.com>
- [7] Web de GitHub [online]: <https://github.com/>
- [8] Web d'Angular [online]: <https://angular.io/>
- [9] Web del material d'Angular [online]: <https://material.angular.io/>
- [10] Components explicats a Angular [online]: <https://material.angular.io/components/categories>
- [11] Tutorial de ShowMyHomework [online]: <https://teamsatchel.wistia.com/medias/37t1f8a6cj?wtime=0>
- [12] Recerques varies: Diferents fòrums de discussió per resoldre dubtes i errors sorgits [online]

APÈNDIXS

APÈNDIX A. MANUAL D'INSTAL·LACIÓ DE L'ENTORN DE DESENVOLUPAMENT

A.1. Dependències

A.1.1. Instal·lació del NODE i NPM

La versió del NODE que es va utilitzar en el desenvolupament es la 8.6.0 així que es recomana descarregar aquesta versió per evitar conflictes entre les versions dels diferents programes utilitzats. Aquests programes ens permetran simular l'entorn de treball del nostre projecte, i sobretot les ordres de l'NPM seran de vital importància cada cop que haguem de treballar amb un nou repositori de l'aplicació o iniciar-ne un de ja instal·lat (aquestes ordres s'expliquen en els següents apartats).

Windows 10 i 8.1:

Per fer-ho, es va a la pàgina web:

<https://nodejs.org/dist/v8.6.0/>

Un cop aquí, es selecciona l'instal·lador corresponent al tipus de sistema de l'ordinador:



node-v8.6.0-linux-s390x.tar.gz	26-Sep-2017 2
node-v8.6.0-linux-s390x.tar.xz	26-Sep-2017 2
node-v8.6.0-linux-x64.tar.gz	26-Sep-2017 2
node-v8.6.0-linux-x64.tar.xz	26-Sep-2017 2
node-v8.6.0-linux-x86.tar.gz	26-Sep-2017 2
node-v8.6.0-linux-x86.tar.xz	26-Sep-2017 2
node-v8.6.0-sunos-x64.tar.gz	26-Sep-2017 2
node-v8.6.0-sunos-x64.tar.xz	26-Sep-2017 2
node-v8.6.0-sunos-x86.tar.gz	26-Sep-2017 2
node-v8.6.0-sunos-x86.tar.xz	26-Sep-2017 2
node-v8.6.0-win-x64.7z	26-Sep-2017 2
node-v8.6.0-win-x64.zip	26-Sep-2017 2
node-v8.6.0-win-x86.7z	26-Sep-2017 2
node-v8.6.0-win-x86.zip	26-Sep-2017 2
node-v8.6.0-x64.msi	26-Sep-2017 2
node-v8.6.0-x86.msi	26-Sep-2017 2
node-v8.6.0.pkg	26-Sep-2017 2
node-v8.6.0.tar.gz	26-Sep-2017 2
node-v8.6.0.tar.xz	26-Sep-2017 2

Figura A.1: Llista d'instal·ladors del node

Ubuntu 17.10:

Una de les maneres d'instal·lar el node en l'Ubuntu, es baixant-se el nodejs tarball de la pàgina oficial en la versió que es desitja (en el nostre cas la 8.6.0), extreure els paquets i fer la instal·lació, o també instal·lar-ho mitjançant el gestor de paquets de Linux (apt).

El NPM s'instal·la quan s'instal·la el NODE. Per comprovar les versions instal·lades en qualsevol sistema operatiu, es fan servir els següents instruccions al terminal del sistema:

```
node -v
v8.6.0
npm -v
5.4.2
```

La versió de l'NPM no és important.

A.1.2. Instal·lació del GIT

El GIT és una eina de treball en comú per a un mateix projecte que permet, amb un sol directori o dependències, ramificar el nostre treball en diferents versions simultànies que es poden guardar i carregar a Github segons les nostres necessitats. D'aquesta manera, diverses persones poden treballar al mateix temps desenvolupant el codi de l'aplicació i un cop finalitzades les funcionalitats per separat es proven i, si funcionen, s'acaben fusionant amb la resta d'avenços i finalment amb la branca mestra del projecte. Tant a l'apartat 3.4. com a l'apèndix ?? hi podreu trobar una sèrie de nocions bàsiques i ordres per poder treballar amb el GIT des de zero tal i com hem hagut de fer en aquest projecte.

Per tal de fer servir les instruccions del github en el terminal del sistema, es necessita instal·lar el programa GIT que es descarrega en aquesta pàgina de web on està l'opció dels sistemes operatius de Windows, Mac OS X i Linux/Unix:

<https://git-scm.com/downloads>

Tot i que la versió del git no es important, per comprovar que s'ha instal·lat i la versió instal·lada es fa servir la instrucció següent en els terminal del sistema:

```
git --version
git version 2.15.0.windows.1
```


A.1.3. Instal·lació del SOURCETREE

El programa sourcetree és un client del Git on es pot veure de manera ràpida els repositoris i les adicions de codi que hagin fet els diferents contribuïdors i contribuïdores.

Aquest programa només està disponible per Windows i Mac OS i s'instal·la amb el següent link:

<https://www.sourcetreeapp.com/>

A.1.4. Instal·lació del VISUAL STUDIO CODE

Per tal de fer els desenvolupaments es recomana el Visual Studio Code ja que és una eina gratuïta, però no és l'única així que es poden fer servir d'altres com el WebStorm de JetBrains. El Visual Studio Code serà l'**editor de codi principal** amb el que treballarem per desenvolupar la nostra aplicació, per tant serà crucial instal·lar-lo i familiaritzar-se amb el llenguatge utilitzat (Java) per tal de poder seguir expandint el projecte.

Per instal·lar el Visual Studio Code pels sistemes operatius de Windows, Linux i Mac OS s'hi va a través del següent link:

<https://code.visualstudio.com/#alt-downloads>

Per tal d'instal·lar-lo en Ubuntu, es fa servir la següent instrucció en el terminal del sistema un cop estem en la carpeta on estigui l'arxiu baixat:

```
sudo dpkg -i nomDeLArchiu.deb
```

Per tal de fer desenvolupaments en HTML5 basats en Typescript i Angular 2 s'han d'instal·lar els següents plugins dins del mateix Visual Studio Code un cop instal·lat:

- Angular 2 plugins:

<https://marketplace.visualstudio.com/items?itemName=johnpapa.Angular2>

- Beutify:

<https://marketplace.visualstudio.com/items?itemName=HookyQR.beautify>

- EditorConfig:

<https://marketplace.visualstudio.com/items?itemName=EditorConfig.EditorConfig>

- Eslint:

<https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>

- SassLint:

<https://marketplace.visualstudio.com/items?itemName=glen-84.sass-lint>

- Tslint:

<https://marketplace.visualstudio.com/items?itemName=eg2.tslint>

- Typescript:

[https://marketplace.visualstudio.com/items?itemName=DSKWRK.
vscode-generate-getter-setter](https://marketplace.visualstudio.com/items?itemName=DSKWRK.vscode-generate-getter-setter)

A.1.5. Instal·lació de l'ANDROID STUDIO

L'Android Studio no és del tot necessària per el projecte, però pot ser útil per debugar l'aplicació d'Android generada per Cordova. En el cas de que no es vulgui instal·lar l'Android Studio, s'ha d'instal·lar l'Android SDK que sí que és necessària.

El programa per les versions de Windows, Mac i Linux es pot trobar en el següent link:

<https://developer.android.com/studio/index.html#download>

A.2. Repositoris

El nostre codi està dividit en tres repositoris diferents: el **services**, el qual es relaciona directament amb l'API i proporciona els mètodes necessaris per a treballar amb cada model fonamental creat; el **dashboard**, que és la pròpia aplicació en sí i ens permet visualitzar tot el codi html, js, json... que es troba relacionat per tal de crear l'entorn de Classpip, i per últim el **mobile**, la versió per a mòbil del dashboard.

Per tal de no modificar el codi de la branca màster haurem de fer el fork de cadascun d'aquests repositoris a l'url del Github corresponent un cop haguem creat el nostre usuari en aquesta pàgina. D'aquesta manera tindrem una còpia 'local' del projecte la qual podrem modificar sense problemes i sense afectar a la resta de branques del projecte.

Si la instal·lació es fa en una màquina amb el Windows com a sistema operatiu, recomanem que abans de començar amb el set up dels tres repositoris, verificar que es tenen alguns components com el Python o el .NET Framework i si no es tenen, instal·lar-los amb la següent instrucció:

```
npm install --global --production windows-build-tools
```

A continuació s'indicaran les instruccions per fer el set-up. Les instruccions que inclouin una **-g** (instal·lació general) és recomanable executar-les amb permisos d'administrador, obrint el cmd del sistema com a administrador. Aquestes instruccions es poden executar des de qualsevol carpeta ja que s'instal·len globalment en el sistema. Les altres instruccions, en canvi, s'han d'executar en les carpetes pertinents dels repositoris clonats. Per tant, haurem d'obrir el cmd a la carpeta corresponent per tal d'executar les ordres en aquesta carpeta. Per fer-ho només cal fer *'Shift + Click dret'* dins la carpeta en qüestió i buscar l'opció *'Abrir ventana de comandos aquí'*.

A.2.1. Classpip-services

<https://github.com/classpip/classpip-services>

1. Fer el Fork del repositori en l'usuari del github (botó a la part superior dreta de la pàgina). Amb això estem copiant al nostre GIT la branca seleccionada per tal de poder treballar amb ella en local sense modificar la que hi ha penjada.
2. Clonar el repositori en local mitjançant la instrucció del terminal o directament baixant el codi en ZIP (i extreure els fitxers). Per comprovar que s'ha baixat correctament, es pot obrir el codi amb el Visual Studio Code o similar.
3. Instal·lar la utilitat Strongloop amb la següent instrucció:

```
npm install -g strongloop@6.0.3
```

Aquesta utilitat serveix per crear entitats i relacions en el codi fàcilment.

En la instal·lació en l'Ubuntu 17.10, entra en bucle en intentar instal·lar una carpeta. Aquest error es deu a que no té accés, marca l'error i ho torna a intentar sense deixar fer res a l'usuari. Es pot parar la instal·lació ja que es una carpeta que no té importància.

En Windows 8.1 es detecten alguns errors, però en executar el servei no dona cap problema.

4. Instal·lar les dependències del projecte (es crearà la carpeta del services a la carpeta seleccionada) executant la següent instrucció en la carpeta del codi font:

```
cd $(HOME)/classpip-services  
npm install
```

(L'ordre cd ens permet passar de les dependències globals del projecte on estàvem executant les ordres fins ara a la carpeta concreta del services sense necessitat d'obrir un nou cmd a la carpeta).

5. Executar el servei Classpip amb la següent instrucció:

```
npm run start
```

S'iniciarà un servidor web local en l'entorn local de la vostra màquina amb els serveis locals publicats. S'obrirà una finestra amb la URL `http://localhost:3000/explorer/` amb una interfície swagger (un framework d'API) on es podran consultar tots els mètodes que s'utilitzen en el codi per relacionar mòduls i models i fer aquesta eina funcional.

És important adonar-se que totes les ordres fins al punt 3 només seran necessàries durant el primer procés d'instal·lació, mentre que a partir del punt 4 trobem les ordres per al cmd local del services que s'hauran d'executar cada cop que obrim el nostre projecte de nou, sense necessitat de tornar a realitzar tots els passos anteriors. A més a més, a partir de la segona vegada que inicialitzem el services, podrem substituir l'ordre **'npm install'** per l'ordre **'node .'**, la qual ens estalviarà temps de compilació.

A.2.2. Classpip-mobile

És molt important saber que la versió en la que es troba el *'mobile'* és més endarrerida que la que presenta el *'dashboard'*, ja que en aquest projecte ens hem centrat en fusionar mòduls i aplicar el major número de funcionalitats possibles tal i com s'ha explicat en apartats anteriors. D'aquesta manera, el *'mobile'* manté la versió desenvolupada al treball de l'**Anna i en Joan** i pot ser posada al dia en futures versions d'aquesta eina (les idees per a futures millores son exposades amb més profunditat a l'apartat ??). De totes maneres, es pot seguir treballant amb aquesta versió i fent proves amb les funcionalitats que incorpora fins al moment, ja que no hem realitzat grans canvis en el *'services'* que afectin de manera directa el funcionament del *'mobile'*.

1. Fer el Fork, clonar i obrir el repositori del següent enllaç tal com s'ha explicat en l'apartat superior:

<https://github.com/classpip/classpip-mobile>

2. Instal·lar a les dependències globals del projecte les següents versions de l'Ionic i el Cordova amb les següents instruccions:

```
npm install -g ionic@2.2.2  
npm install -g cordova@6.4.0
```

3. Instal·lar les dependències locals per l'aplicació mòbil amb les següents instruccions:

```
cd ${HOME}/classpip-mobile  
npm install  
mkdir www  
cordova prepare
```

En la instal·lació en Ubuntu 17.10, en fer la instrucció cordova prepare, salta un error que diu que no troba el plugin mx.ferreyra.callnumber que s'hauria de trobar en el fitxer config.xml. Aquest plugin serveix per trucar directament als números de telèfon escrits a l'aplicació sense haver de sortir d'aquesta.

Si s'esborra el plugin, es pot continuar amb la instal·lació sense problemes (no es tindrà, per tant la funcionalitat de trucar des de l'aplicació). En un futur es canviarà el plugin per l'actual.

Per a provar l'aplicació s'ha d'executar la següent instrucció:

```
ionic serve
```

Si es vol veure l'aplicació en format mòbil al navegador s'ha d'executar la següent instrucció:

```
ionic serve --lab
```

Hi ha dos entorns per desenvolupar, l'entorn de producció i el de desenvolupament. El de desenvolupament, tal com indica el seu nom, és per fer proves i començar a desenvolupar i, una vegada funciona la part de codi que es vol fer, es puja a producció.

Altres cops, tal i com hem comentat abans, únicament és necessari utilitzar les ordres de 'npm install' i 'ionic serve' a partir de la segona execució del programa, i no és necessari repetir la resta d'ordres per fer funcionar la versió del mòbil.

Per tal de començara fer una petita prova amb el codi i programar alguna cosa de l'aplicació en un servei local (al *localhost:3000* en comptes de remot (el qual a priori no funcionarà), s'ha de comentar i descomentar les següents línies del fitxer que trobareu al directori següent del projecte:

/classpip-mobile/src/app/app.config.ts

```
// public static get SERVER_URL(): string { return 'https://api.classpip.com'; } // PRO  
public static get SERVER_URL(): string { return 'http://localhost:3000'; } // DEV
```

A.2.3. Classpip-dashboard

1. Fer el Fork, clonar i obrir el repositori del següent enllaç tal com s'ha explicat en l'apartat superior:

<https://github.com/classpip/classpip-dashboard>

2. Instal·lar les dependències locals a la carpeta del classpip-dashboard amb les següents instruccions:

```
cd ${HOME}/classpip-dashboard  
npm install
```

3. Per executar l'aplicació de dashboard s'utilitza la següent instrucció:

```
npm run start
```

Per obrir l'aplicació, s'ha d'anar al navegador i anar a la següent adreça:

<http://localhost:4200/>

En l'entorn del dashboard serà on es desenvoluparà majorment el nostre projecte, la interconnexió de mòduls i la integració de millores als mòduls existents i noves funcionalitats (sent aplicables més tard a la versió mòbil).

APÈNDIX B. TUTORIAL MESA

L'objecte, els mètodes i la pàgina que es creen en aquest manual no es troben en el projecte ja que considerem que no compleixen cap necessitat ni serveixen d'utilitat actualment per a l'eina que s'està desenvolupant. És important saber que, com aquest procés d'aprenentatge no tindrà un impacte directe en les funcionalitats del projecte ja desenvolupat, aquests passos purament educacionals es poden dur a terme tant en les versions més recents com en d'altres més antigues. Recomanem utilitzar els repositoris de la versió de **l'Anna i en Joan (resultat) partint dels d'en Ferran** ja que el tutorial està adaptat per a aquesta versió i la comparativa obtinguda en el procés d'aprenentatge serà més visual i directa, però si més endavant us hi atreviu podeu agafar els repositoris més recents i intentar replicar-hi aquests resultats. D'aquesta manera, a continuació es mostren els repositoris amb el resultat final obtingut en seguir aquest tutorial, per tal de poder comparar-los amb el que l'estudiant estigui duent a terme:

Repositoris del resultat final d'aquest tutorial a mode comparatiu (versió Anna i Joan):

- **Classpip-services:**

<https://github.com/gazorpAnna/classpip-services-tutorial.git>

- **Classpip-mobile:**

<https://github.com/gazorpAnna/classpip-mobile-tutorial.git>

- **Classpip-dashboard:**

<https://github.com/gazorpAnna/classpip-dashboard-tutorial.git>

B.1. Repositoris de partida

Aquesta guia parteix dels repositoris de Serveis, de Mobile i de Dashboard de la versió d'en **Ferran**, la qual és com un esquelet o base del projecte sense cap mòdul instal·lat. Aquesta versió més senzilla serà més indicada que les més recents per familiaritzar-se amb el codi i el llenguatge de programació gràcies a la inferior quantitat de línies de codi i interrelacions entre mòduls.

Per tal de reproduir l'entorn en que es basa aquesta guia mitjançant el gitHub, s'han de descarregar els següents repositoris:

Repositoris inicials d'aquest tutorial (versió Ferran):

- **Classpip-services:**

<https://github.com/classpip/classpip-services>

- **Classpip-mobile:**

<https://github.com/classpip/classpip-mobile>

- **Classpip-dashboard:**

<https://github.com/classpip/classpip-dashboard>

B.2. Objectiu i definició de l'objecte

L'objectiu es crear un nou objecte “**Mesa**” que tindrà la informació de la taula a on estigui l'estudiant que utilitzi l'aplicació. Per tant, només podrà veure aquesta informació un usuari que tingui un rol d'estudiant.

Atributs de l'objecte:

- **Id:** per identificar el registre.
- **nombreAula:** nom de l'aula on és la taula.
- **numFila:** número de la fila de taules de l'aula.
- **numColumna:** número de la columna de taules de l'aula.
- **imageM:** imatge de la taula.

En l'aplicació mòbil, es podrà accedir a la pàgina de “Mesa” mitjançant la pàgina “home”, on es mostrarà l'atribut de *nombreAula* i des del menú lateral de l'aplicació, on es mostrarà la variable TITLE de l'objecte.

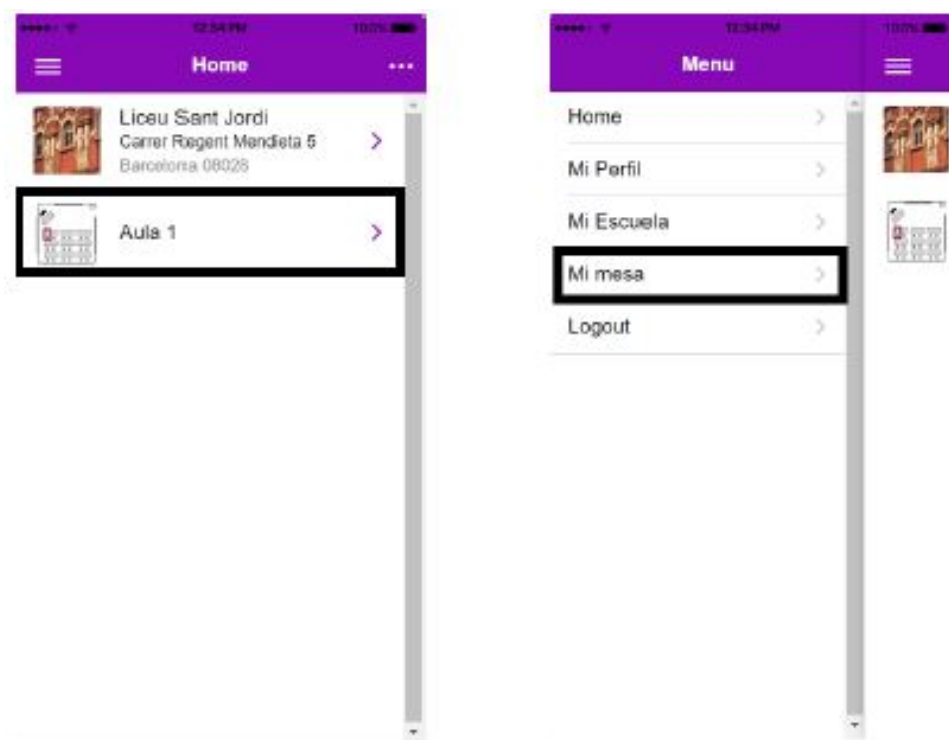


Figura B.1: Visualització dels punts de menú del resultat final del manual

I la visualització de la pàgina serà:

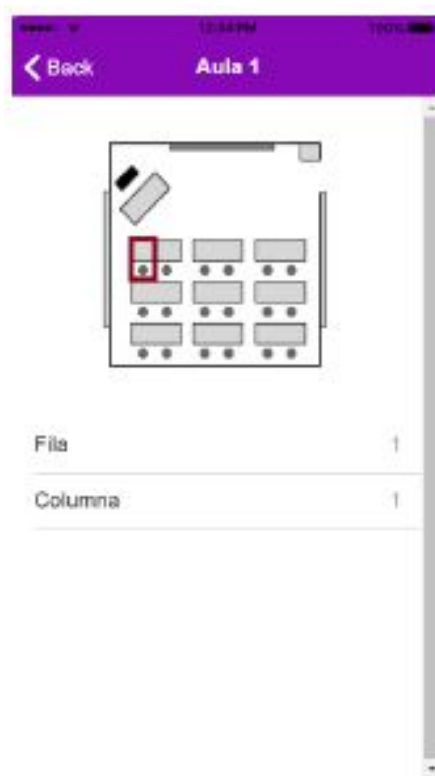


Figura B.2: Visualització de la pàgina de menú del resultat final del manual de la versió mobile

En quant a la pàgina web de la versió dashboard, es podrà accedir mitjançant la barra de navegació superior, on hi haurà un botó que portarà a la pàgina que mostrarà la informació de “Mesa”:



Figura B.3: Visualització de la pàgina de menú del resultat final del manual de la versió dashboard

B.3. Passos a seguir per dur a terme el tutorial

B.3.1. Creació del model 'Mesa' i les dades al repositori Services

L'API de l'aplicació es crea automàticament mitjançant la definició del model de dades i les relacions entre els seus objectes.

Per tal de crear el model de dades per enviar i rebre les peticions de la API, s'han de crear dos fitxers amb el mateix nom (en el nostre cas els anomenem "Mesa"), un amb format **js** (javascript) i l'altre en format **json** que es on estarà la definició de les dades, les relacions i el nivell d'accés que es podrà tenir a aquestes dades.

Per simplificar, l'objecte "Mesa" no tindrà cap relació amb cap altre objecte creat, únicament serà l'estudiant el qual tindrà una taula assignada.

El model, per tal de seguir l'estructura del projecte, s'haurà de crear a la carpeta *common* → *models*, allotjada dins el repositori de **services**.

- **Model 'Mesa':**

Es crea el fitxer en format js i simplement s'afegeixen les següents línies que serveixen per definir el nom de la funció:

```
1      'use strict';
2
3      /* eslint-disable */
4      module.exports = function (Mesa) {};
5      /* eslint-enable */
6
```

Figura B.4: mesa.js

En crear el fitxer en format json s'han d'afegir el nom i les propietats de l'objecte, les relacions i el nivell d'accés segons el rol (estudiant o professor):

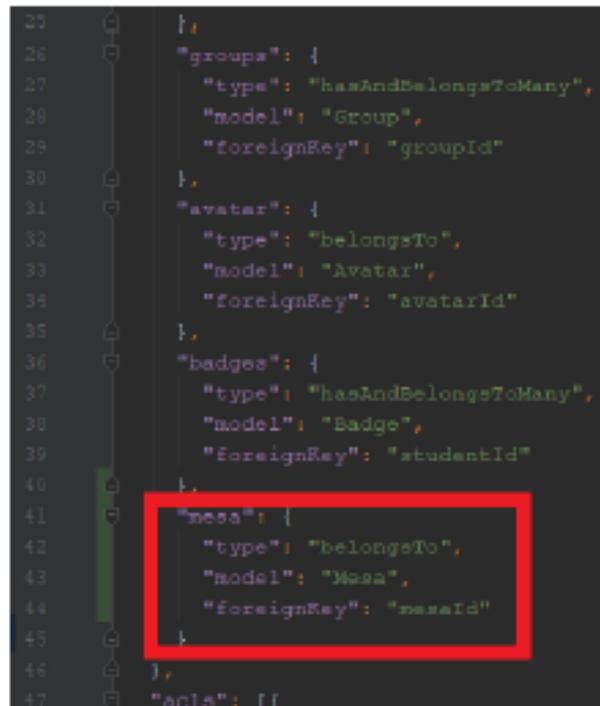
```
1 {
2   "name": "Mesa",
3   "plural": "mesas",
4   "base": "PersistedModel",
5   "idInjection": true,
6   "options": {
7     "validateUpsert": true
8   },
9   "properties": {
10    "nombreAula": {
11      "type": "string",
12      "required": true
13    },
14    "numFila": {
15      "type": "number",
16      "required": true
17    },
18    "numColumna": {
19      "type": "number",
20      "required": true
21    },
22    "imageM": {
23      "type": "string",
24      "required": true
25    }
26  },
27  "validations": [],
28  "relations": {
29  },
30  },
31  "acls": [
32    {
33      "accessType": "*",
34      "principalType": "ROLE",
35      "principalId": "$everyone",
36      "permission": "DENY"
37    },
38    {
39      "accessType": "READ",
40      "principalType": "ROLE",
41      "principalId": "$everyone",
42      "permission": "ALLOW"
43    },
44    {
45      "accessType": "WRITE",
46      "principalType": "ROLE",
47      "principalId": "SYS-ADMIN",
48      "permission": "ALLOW"
49    }
50  ],
51  "methods": {}
52 }
```

Figura B.5: mesa.json

L'Id no cal que s'especifiqui ja que no es una propietat en si de l'objecte, si no que es una propietat que serveix per relacionar objectes i que l'API ja dona per suposada.

- **Model 'Student':**

Per tal de lligar un estudiant amb la seva taula, s'ha d'afegir la relació en el model de dades de l'estudiant (arxiu *student.json* a la mateixa carpeta de models):



```
25  {
26    "groups": {
27      "type": "hasAndBelongsToMany",
28      "model": "Group",
29      "foreignKey": "groupId"
30    },
31    "avatar": {
32      "type": "belongsTo",
33      "model": "Avatar",
34      "foreignKey": "avatarId"
35    },
36    "badges": {
37      "type": "hasAndBelongsToMany",
38      "model": "Badge",
39      "foreignKey": "studentId"
40    },
41    "messa": {
42      "type": "belongsTo",
43      "model": "Messa",
44      "foreignKey": "messaId"
45    },
46  },
47  "acla": []
```

Figura B.6: student.json

Arribats a aquest punt és imprescindible mencionar la manera amb la qual relacionarem diferents models o objectes en aquest projecte. Per tal de lligar dos objectes diferents haurem d'establir la relació entre ells, que pot ser dels següents tipus:

- **'belongsTo'**: Relació establerta quan un objecte de tipus 'A' pertany a un objecte de tipus 'B' (unidireccionalment). Per exemple, un grup sencer de classe pertany a un professor però no a la inversa, tal i com estava plantejat a la versió inicial del mòdul de competicions previ a la integració en el nostre projecte.
- **'hasMany'**: Relació establerta quan un objecte de tipus 'B' té lligats molts objectes de tipus 'A' (unidireccionalment). Per exemple, en el cas anterior podem dir que un professor té diversos grups al seu càrrec.
- **'hasAndBelongsToMany'**: Relació bidireccional que estableix que un objecte de tipus 'A' es troba relacionat amb molts objectes de tipus 'B' i exactament igual a la inversa. Un bon exemple seria un objecte de tipus grup de classe que té molts objectes de tipus student assignats i alhora aquest grup també pertany a aquests alumnes, sent també cert que els students pertanyen a aquest grup.

La relació, en el nostre cas serà del tipus “BelongsTo”, és a dir, que cada estudiant té una sola taula assignada i utilitzarem com a clau primària “mesald”.

Una vegada acabats els models s’ha d’afegir l’objecte al fitxer **model-config.json** i especificar la base de dades i si és un objecte públic o no:

```
36     },
37     "Mesa": {
38         "dataSource": "db",
39         "public": true
40     },
41     "student": {
```

Figura B.7: config.json

B.3.2. Visualització a l’API

Una vegada completats els models i la definició del model, podem comprovar que s’ha creat correctament l’objecte a la API executant el mòdul de services i obrint la pàgina en el port especificat al final de la secció [A.2.1.](#):

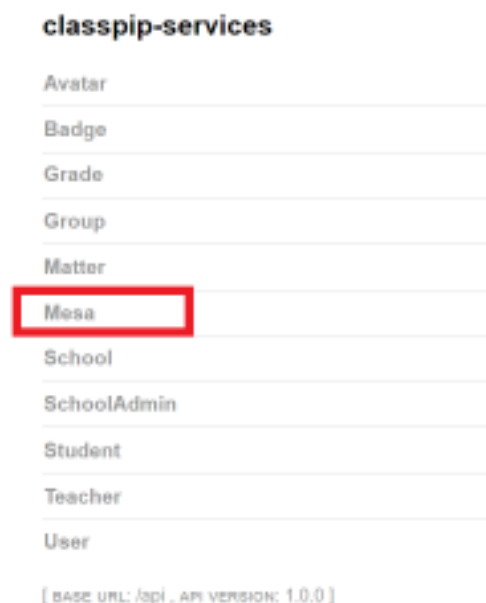


Figura B.8: Visualització de l’objecte a l’API

Com podem comprovar s’ha creat l’objecte i si despleguem els mètodes podrem veure els que es creen automàticament per defecte ja que no hem assignat cap relació a l’objecte:

Group	Show/Hide	List Operations	Exposed Operations
Matter	Show/Hide	List Operations	Exposed Operations
Mesa	Show/Hide	List Operations	Exposed Operations
PATCH	/mesas	Patch an existing model instance or insert a new one into the data source.	
GET	/mesas	Find all instances of the model matched by filter from the data source.	
PUT	/mesas	Patch an existing model instance or insert a new one into the data source.	
POST	/mesas	Create a new instance of the model and persist it into the data source.	
PATCH	/mesas/{id}	Patch attributes for a model instance and persist it into the data source.	
GET	/mesas/{id}	Find a model instance by {id} from the data source.	
HEAD	/mesas/{id}	Check whether a model instance exists in the data source.	
PUT	/mesas/{id}	Patch attributes for a model instance and persist it into the data source.	
DELETE	/mesas/{id}	Delete a model instance by {id} from the data source.	
GET	/mesas/{id}/nodes	Check whether a model instance exists in the data source.	
POST	/mesas/{id}/replace	Replace attributes for a model instance and persist it into the data source.	
GET	/mesas/change-stream	Create a change stream.	
POST	/mesas/change-stream	Create a change stream.	

Figura B.9: Visualització de les funcionalitats de l’API

En canvi, si despleguem l’objecte “Students” podrem trobar la definició de la url que correspon a la relació que hem afegit al model:

PUT	/students/{id}/groups/rel/{fk}	Add a related item by id for groups.
DELETE	/students/{id}/groups/rel/{fk}	Remove the groups relation to an item by id.
GET	/students/{id}/mesa	Fetches belongsTo relation mesa.
POST	/students/{id}/replace	Replace attributes for a model instance and persist it into the data source.
GET	/students/{id}/school	Fetches belongsTo relation school.

Figura B.10: Visualització de les funcions creades en l’objecte student

B.3.3. Creació de les dades inicials

Ja que no executem el programa des del servidor de producció, no estem connectats a una base de dades que contingui dades per poder provar que funciona el que estem programant (estem treballant en adreces locals).

Per tant, per poder visualitzar-les a l'aplicació, hem de crear un fitxer a la carpeta *server* → *boot* del repositori *services* amb la informació següent:

```
1  'use strict';
2
3  module.exports = function (app, cb) {
4
5    app.models.Mesa.create([
6      {
7        nombreAula: 'Aula 1',
8        numFile: 1,
9        numColumna: 1,
10       imageM: 'assets/mesa1.png'
11     }, {
12       nombreAula: 'Aula 1',
13       numFile: 2,
14       numColumna: 4,
15       imageM: 'assets/mesa3.png'
16     }, {
17       nombreAula: 'Aula 2',
18       numFile: 1,
19       numColumna: 6,
20       imageM: 'assets/mesa2.png'
21     }
22   ], function (err) {
23     if (err) throw err;
24     process.nextTick(cb);
25   });
26 }
```

Figura B.11: Nou arxiu 01-01-create-mesa.js

Així que, la ruta de les imatges que s'ha d'indicar en la creació de les dades es el nom de la carpeta + el nom de la imatge.

Les imatges es poden declarar de dues maneres en el codi per tal d'afegir-les a la nostra aplicació:

- Definint la URL d'un lloc d'internet.
- Definint la ruta de la carpeta on està guardada la imatge. Aquest mètode només es pot considerar si la imatge ha sigut creada i no es troba a internet.

En el cas de les imatges de la taula, es troben en la carpeta *assets* del repositori del Mobile. Per tal d'utilitzar les mateixes que al resultat final, amb el mateix format d'imatge i dimensions, extraieu-les de la carpeta esmentada però del projecte amb la Mesa funcional (l'utilitzat per comparar si esteu realitzant correctament els passos d'aquest tutorial) i afegiu-les a la mateixa carpeta però del projecte que esteu creant amb el tutorial:

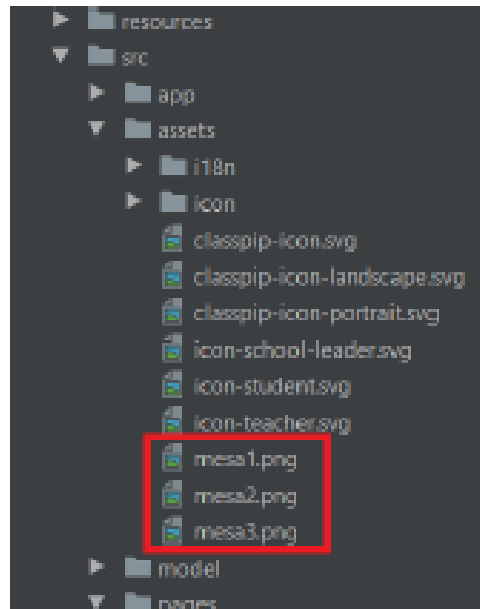


Figura B.12: Carpeta d'assets del repositori Mobile

Un cop copiades les imatges a la carpeta adequada i juntament amb el codi de l'arxiu 01 – 01 – *create – mesa.js*, haureu aconseguit crear una funció que més endavant us permetrà visualitzar les taules a l'aplicació. D'aquesta manera, en aquest cas particular tindrem tres taules a visualitzar.

A part de crear les taules, les hem d'assignar als estudiants en el fitxer 06 – *create – students.js* mitjançant la clau especificada en les relacions del model Students: “mesald”, amb la qual assignarem a cada estudiant una taula en particular:

```
module.exports = function (app, db) {  
  
  app.model.Student.create([  
    {  
      id: 10000,  
      username: 'student-1',  
      password: 'student-1',  
      name: 'Lorena',  
      surname: 'Diaz',  
      email: 'student-1@classpip.com',  
      emailVerified: 1,  
      schoolId: 1,  
      avatarId: 6,  
      mesald: 1  
    }, {  
      id: 10001,  
      username: 'student-2',  
      password: 'student-2',  
      name: 'Roxario',  
      surname: 'Arrellano',  
      email: 'student-2@classpip.com',  
      emailVerified: 1,  
      schoolId: 1,  
      avatarId: 6,  
      mesald: 2  
    }, {  
      id: 10002,  
      username: 'student-3',  
      password: 'student-3',  
      name: 'Gillermo',  
      surname: 'Macho',  
      email: 'student-3@classpip.com',  
      emailVerified: 1,  
      schoolId: 2,  
      avatarId: 7,  
      mesald: 3  
    }  
  ])
```

Figura B.13: Arxiu 06-create-students.js

Una vegada completats tots els passos, es pot executar la part de Services i comprovar tant a l'API com amb el *Postman* que es porti la informació correcta (l'explicació del funcionament del *Postman* es fa en una secció més endavant).

Les imatges que es defineixen en les dades inicials s'han afegit als assets dels repositoris de Mobile i de Dashboard, però es poden utilitzar imatges d'internet indicant la url en l'atribut.

B.3.4. Verificació de les dades mitjançant Postman i l'API (pas opcional)

Per tal de fer crides a l'API mitjançant el programa *Postman* (cal descarregar-lo d'internet) o l'API (entrant al *localhost* : 3000 tal i com s'ha especificat en anteriors apartats) en si, s'ha de tenir en compte la capçalera de la petició, que es correspon al **token**, el qual és una clau que es crea en autenticar-se amb un usuari introduint l'usuari i la contrasenya en l'aplicació.

D'aquesta manera, si el programa es divideix en diversos servidors només caldrà que els usuaris i les contrasenyes estiguin guardats en un, i que s'accedeixi als altres mitjançant aquesta clau. Aquest fet incrementa molt la seguretat pel fet que només s'ha de protegir les contrasenyes i noms d'usuari en un únic servidor.

- Com aconseguir el Token:

Quan s'accedeix a l'aplicació introduint les credencials, es genera automàticament un token que es guarda en la variable **currentUser.id** del provider *utils.services.ts*. S'ha de tenir en compte de que cada vegada que es fa un login, es canvia el token, de manera que si canviem d'usuari, es generarà un de nou.

Una manera d'aconseguir-ho seria fent una petita modificació del codi perquè en alguna pàgina de l'aplicació mòbil o web es mostri i es pugui copiar.

En aquesta guia no s'explicarà com fer aquesta modificació, però sí que està feta als repositoris amb el codi de la guia, en l'aplicació mòbil, en la pàgina del perfil:

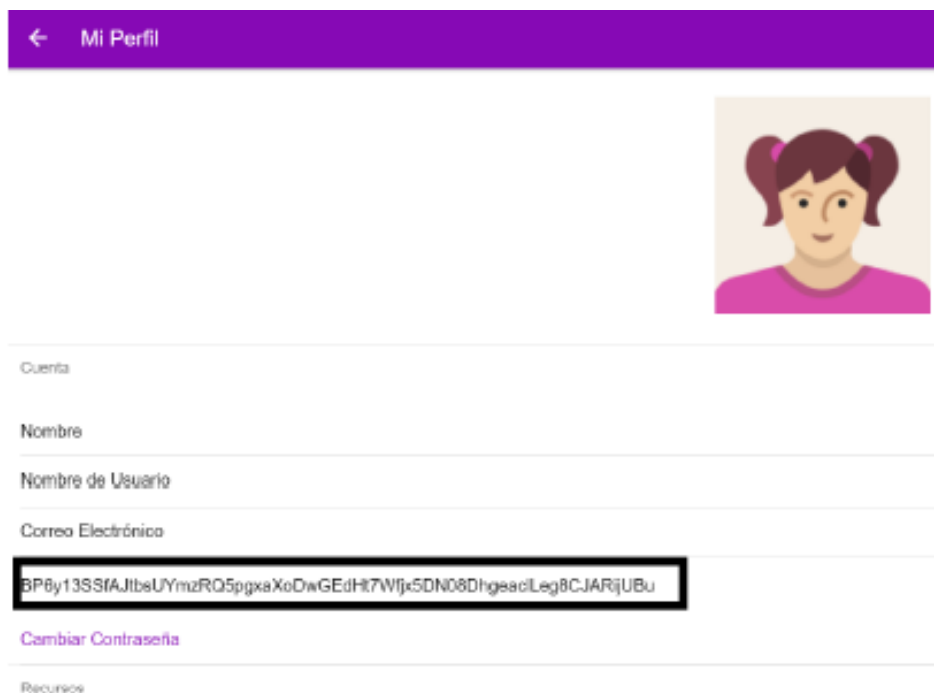


Figura B.14: Visualització del token en el perfil

- Afegir el token al Postman:

Per afegir el token al programa del Postman que ja haureu descarregat prèviament d'internet, en la pestanya “Headers” s’ha d’afegir una línia d’autenticació copiant el token en la part de Values:

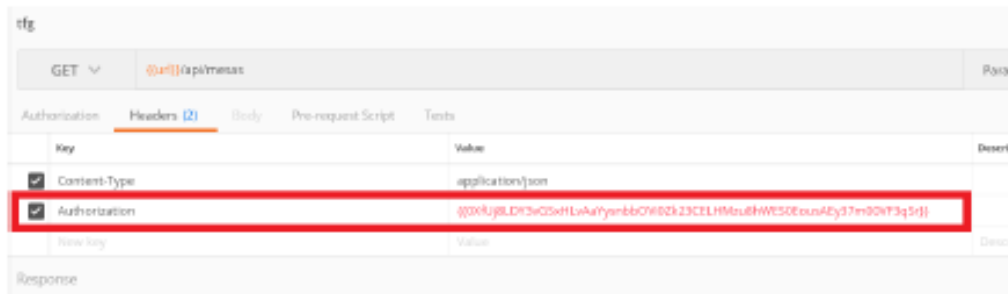


Figura B.15: Lloc on introduir el token al Postman

- Afegir el token a l'API:

En l'API, simplement s'ha de copiar el token a la part superior i pressionar el botó:

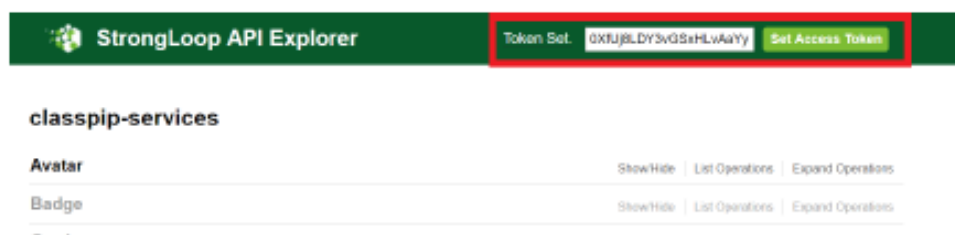


Figura B.16: Lloc on introduir el token a StrongLoop (l'API)

B.3.5. Creació de les pàgines al repositori Mobile

Per tal de començar aquest pas és important saber el següent: quan es parla d'un objecte seguit del terme “Page”, ens referirem a la pàgina corresponent del directori *src* \mapsto *pages* que es troba dins la carpeta del repositori Mobile.

Per exemple, si obrim la carpeta de la pàgina del *Menu*, hi trobarem tres arxius diferents (*l'HTML, l'SCSS i el TS*), però quan parlem de *MenuPage* ens estem referint concretament al que està creant la classe d'aquesta pàgina, és a dir, **l'arxiu TS**:

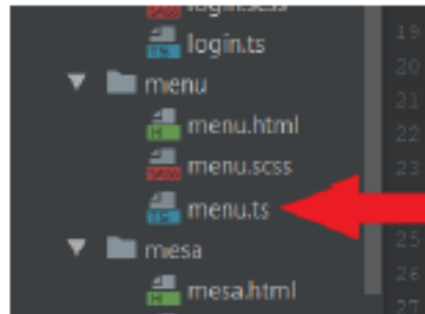


Figura B.17: Visualització dels tres arxius del directori de la pàgina 'mesa'

El primer pas és la creació del model per tal de poder rebre dades de la API i convertir-les en l'objecte "Mesa" en si mitjançant les funcions, models i relacions que hem especificat en els anteriors passos (traslladem la feina anterior al Mobile per poder-la utilitzar aquí). Per fer-ho, crearem una classe del tipus **typescript (TS)** dins de la carpeta *src* \mapsto *models* i especificarem els atributs de l'objecte, el constructor, el mètode (**toObject**) per passar a l'objecte "Mesa" l'objecte genèric que es rep i els corresponents gets i sets segons els atributs especificats:

```
1  export class Mesa {
2
3      private _id: string;
4      private _nombreAula: string;
5      private _numFila: number;
6      private _numColumna: number;
7      private _imageM: string;
8
9
10     constructor(id?: string, nombreAula?: string, numFila?: number, numColumna?: number, imageM?: string) {
11         this._id = id;
12         this._nombreAula = nombreAula;
13         this._numFila = numFila;
14         this._numColumna = numColumna;
15         this._imageM = imageM;
16     }
17     /* tslint:disable */
18     static toObject(object: any): Mesa {
19         /* tslint:enable */
20         let result: Mesa = new Mesa();
21         if (object != null) {
22             result.id = object.id;
23             result.nombreAula = object.nombreAula;
24             result.numFila = object.numFila;
25             result.numColumna = object.numColumna;
26             result.imageM = object.imageM;
27         }
28         return result;
29     }
30 }
```

Figura B.18: Classe del model Mesa creada al nou arxiu *mesa.ts* en el directori especificat

També cal afegir l'atribut “mesald” al model “Student” (altre cop a la carpeta *src* \mapsto *model*) per tal que es tradueixi correctament (i els seus corresponents get i set):

```
5 private _id: string;
6 private _name: string;
7 private _surname: string;
8 private _username: string;
9 private _email: string;
10 private _schoolId: number;
11 private _avatarId: number;
12 private _avatar: Avatar;
13 private _mesaId: number;
14
15 constructor(name?: string, surname?: string, username?: string,
16 email?: string, schoolId?: number, avatarId?: number, mesaId?: number) {
17   this._name = name;
18   this._surname = surname;
19   this._username = username;
20   this._email = email;
21   this._schoolId = schoolId;
22   this._avatarId = avatarId;
23   this._mesaId = mesaId;
24 }
25
26 /* tslint:disable */
27 static toObject(object: any): Student {
28   /* tslint:enable */
29   let result: Student = new Student();
30   if (object != null) {
31     result.id = object.id;
32     result.name = object.name;
33     result.surname = object.surname;
34     result.username = object.username;
35     result.email = object.email;
36     result.schoolId = object.schoolId;
37     result.avatarId = object.avatarId;
38     result.mesaId = object.mesaId;
39   }
40   return result;
41 }
```

Figura B.19: Modificació del model “Student” a *student.ts*

Una vegada creat el model Mesa, creem els mètodes que generen peticions a l'API dins de la carpeta *src* \mapsto *providers*. En el nostre cas, només necessitarem un mètode que l'anomenarem “**getMyMesa**”, el qual retornarà l'objecte sencer:

```
1 import {Injectable} from "@angular/core";
2
3
4
5
6
7
8
9
10 @Injectable()
11 export class MesaService {
12
13   constructor(
14     public http: Http,
15     public utilsService: UtilsService) { }
16
17   public getMyMesa(): Observable<Mesa> {
18
19     let options: RequestOptions = new RequestOptions({
20       headers: this.utilsService.setAuthorizationHeader(new Headers(), this.utilsService.currentUser.id)
21     });
22
23     var url: string = this.utilsService.getMyUrl() + AppConfig.MYMESA_URL;
24
25     return this.http.get(url, options)
26       .map((response: Response, index: number) => Mesa.toObject(response.json()))
27       .catch((error: Response) => this.utilsService.handleAPIError(error));
28   }
29 }
```

Figura B.20: Nou arxiu provider anomenat *mesa.service.ts* creat al directori especificat

Les url de les peticions es fixen en un fitxer a part dins de la carpeta *src* \mapsto *app* amb el nom de **app.config.ts**.

Per saber quina url es necessita, simplement cal consultar els mètodes generats a l'API i l'únic que ens interessa de moment és el de la figura següent, ja que ens retorna la “Mesa” lligada a un estudiant.

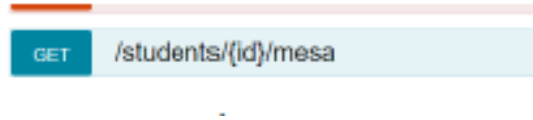


Figura B.21: Mètode 'get mesa' de l'API

Per tant, només haurérem d'afegir la part de “/mesa” a l'arxiu que hem esmentat, ja que el mètode “**getMyUrl**” ja ens retorna la part de l'estudiant i els seu id:

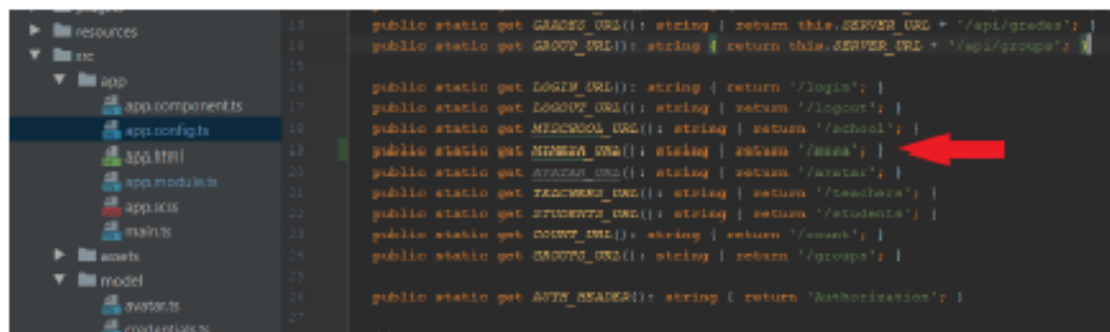


Figura B.22: Definició de la ruta dels mètodes de l'API de l'objecte mesa

L'objecte “Mesa” s'inicialitzarà tant a la pàgina del menú com a la pàgina home ja que es podrà accedir per les dues pàgines.

Tan en la MenuPage com en la HomePage hi haurà un mètode “vesAlaMesa” on dins hi haurà la petició “getMyMesa()” que hem creat anteriorment en el mesaService per tal de poder retornar la taula adequada quan ho demanem per l'aplicació.

Dins de la petició, si es rep la informació correctament i sense errors, s'utilitza la classe *NavController* per inicialitzar la pàgina *MesaPage*, passant-li la variable “mesa” inicialitzada i amb informació.

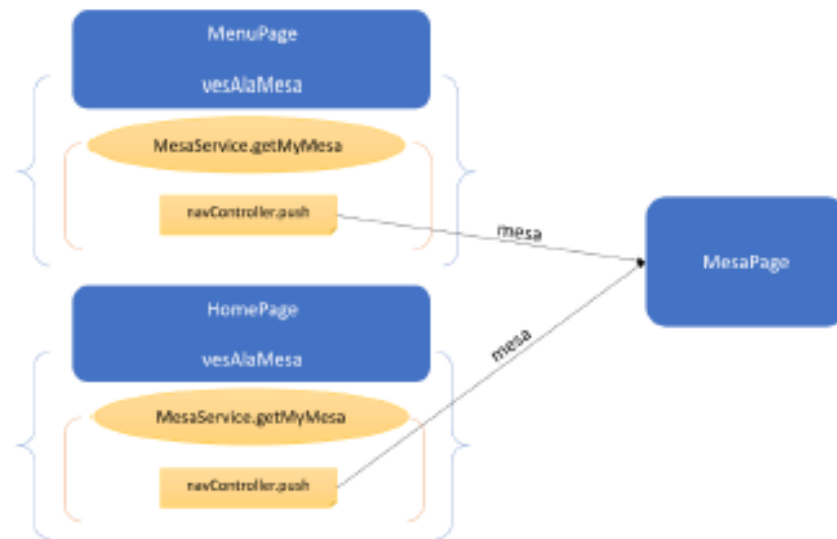


Figura B.23: Esquema explicatiu de l'accés a la pàgina de mesa

Per tant, per fer la crida des de la *MenuPage*, es defineixen les variables que s'utilitzen i s'inicialitza la *MesaPage* (cal anar al fitxer *menu.ts*, al directori *src* \mapsto *pages* \mapsto *menu* i afegir les següents línies de codi):

```

24 export class MenuPage {
25
26   @ViewChild(Nav) nav: Nav;
27
28   public rootPage: Component;
29   public homePage: Page;
30   public schoolPage: Page;
31   public mesaPage: Page;
32
33   public myRole: Role;
34   public role = Role;
35
36   constructor(
37     public navCtrl: NavController,
38     public translateService: TranslateService,
39     public utilsService: UtilsService,
40     public ionicService: IonicService,
41     public schoolService: SchoolService,
42     public mesaService: MesaService,
43     private loginService: LoginService) {
44
45     this.rootPage = HomePage;
46     this.homePage = new Page(HomePage, this.translateService.instant({ key: 'HOME.TITLE'}));
47     this.schoolPage = new Page(SchoolPage, this.translateService.instant({ key: 'SCHOOL.TITLE'}));
48     this.mesaPage = new Page(MesaPage, this.translateService.instant({ key: 'MESA.TITLE'}));
49     this.myRole = this.utilsService.role;
50   }

```

Figura B.24: Punt d'entrada a la *MesaPage* des de *MenuPage* a l'arxiu *menu.ts*

I després el mètode en si té una forma:

```
92 | public vesAlaMesa(): void {  
93 |  
94 |     this.ionicService.showLoading(this.translateService.instant('APP.WAIT'));  
95 |  
96 |     this.mesaService.getMyMesa().subscribe(  
97 |         ((value: Mesa) => this.navController.push(MesaPage, { mesa: value })),  
98 |         error => {  
99 |             this.ionicService.showAlert(this.translateService.instant('APP.ERROR'), error);  
100 |             this.ionicService.removeLoading();  
101 |         });  
102 | }
```

Figura B.25: Mètode que conté el direccionament a la MesaPage (també a *menu.ts*)

També s'ha de definir el botó a *menu.html* posant com a condició que el rol de l'usuari que podrà veure aquesta informació sigui el d'**estudiant**. Si no es un estudiant, no es mostrarà. Les següents línies hauran de ser afegides a l'arxiu *menu.html* (ja que a l'HTML es on programarem tota la interfície visual de la nostra aplicació), dins el directori *src* \mapsto *pages* \mapsto *menu*:

```
17 |  
18 | <button menuClose ion-item *ngIf="myRole == role.STUDENT" (click)="vesAlaMesa()">  
19 |     {{mesaPage.title}}  
20 | </button>  
21 | <button menuClose ion-item (click)="logout()">
```

Figura B.26: Botó que crida al mètode de direccionament a la MesaPage

Pel que respecta a la part de la HomePage és igual que la de MenuPage, l'única part que canvia és la que es mostra d'entrada a la HomePage:

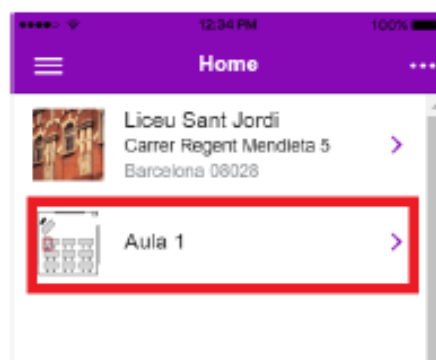


Figura B.27: Visualització del botó de direccionament

La informació del botó s'ha de generar en el moment que s'inicialitza la pàgina, és a dir, dins del mètode “**getHomeInfo()**” que trobareu a l'arxiu *home.ts* al directori *src* \mapsto *pages* \mapsto *home*, i com que només serà visible per als estudiants, ha d'estar dins de l'If que correspongui als estudiants:

```
110 } else if (this.myRole === Role.STUDENT) {
111
112   this.schoolService.getMySchool().finally(() => {
113     refresher ? refresher.complete() : null;
114     this.ionicService.removeLoading();
115   }).subscribe(
116     ((value: School) => {
117       this.school = value;
118
119       this.mesaService.getMyMesa().subscribe(
120         ((value: Mesa) => {
121           this.mesa = value
122         }),
123         error => this.ionicService.showAlert(this.translateService.instant('APP.ERROR'), error));
124       }
125     ),
126     error => this.ionicService.showAlert(this.translateService.instant('APP.ERROR'), error));
127
128   }
129 }
130 }
```

Figura B.28: Mètode *getter* de la taula de l'estudiant

I el botó programat en el fitxer *home.html* tindrà la forma:

```
30 </ion-item>
31 <ion-item *ngIf="mesa" (click)="vesAlaMesa()">
32   <ion-thumbnail item-left>
33     <img [src]="mesa.imageM">
34   </ion-thumbnail>
35   <h2>{{mesa.nombreAula}}</h2>
36   <button ion-button clear item-right icon-only>
37     <ion-icon name="ios-arrow-forward"></ion-icon>
38   </button>
39 </ion-item>
40 </ion-list>
41 <!-- TEACHERS -->
```

Figura B.29: Botó de direccionament a la MesaPage en el menú Home

Una vegada creats els models, els mètodes que interaccionen amb l'API i els mètodes que porten a la MesaPage, es procedeix a crear la pàgina “mesa” dins de la carpeta *src* \mapsto *pages*, on, per seguir l'estructura del projecte, s'haurà de crear un directori i dins els tres fitxers corresponents:

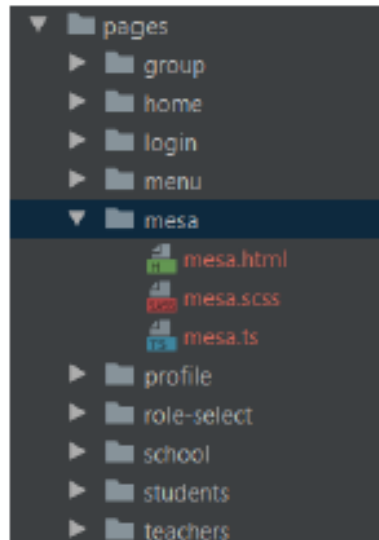


Figura B.30: Fitxers de la pàgina de la taula

A continuació s'explica com crear correctament aquests arxius per garantir la seva funcionalitat:

- **Mesa.ts:**

```
1  import ...
6
7  @Component({
8    selector: 'page-mesa',
9    templateUrl: './mesa.html'
10 })
11
12 export class MesaPage {
13
14   public mesa: Mesa;
15
16   constructor(
17     public navParams: NavParams,
18     public translateService: TranslateService,
19     public ionicService: IonicService) {
20
21     this.mesa = this.navParams.data.mesa;
22   }
23   public ionViewDidEnter(): void {
24     this.ionicService.removeLoading();
25   }
26 }
27
28 }
```

Figura B.31: Nou arxiu *mesa.ts*

Aquest fitxer correspon a les funcions que fa la pàgina. Com que en el nostre cas només es tracta d'una pàgina que ensenya informació i l'objecte s'ha inicialitzat a les pàgines anteriors, només s'ha de recuperar la informació.

A més a més, serà important haver afegit els **imports** necessaris per poder implementar totes les funcionalitats d'aquest arxiu:

```
1 import {Mesa} from "../../model/mesa";
2 import {TranslateService} from "ng2-translate/ng2-translate";
3 import {IonicService} from "../../providers/ionic.service";
4 import {NavParams} from "ionic-angular";
5 import {Component} from "@angular/core";
```

Figura B.32: *imports* necessaris a l'arxiu *mesa.ts*

- **Mesa.scss:**

En aquest document es defineix l'estil dels elements de la pàgina. Cal esmentar que quan en un arxiu *scss* es parli de '**px**' estarem definint la quantitat de **píxels** que estem deixant de marge, segons el terme utilitzat:

```
1 .page-mesa {
2   .logo-row {
3     padding-bottom: 20px;
4     padding-top: 20px;
5   }
6
7   ion-row {
8     align-items: center;
9     text-align: center;
10  }
11 }
12
```

Figura B.33: Nou arxiu *mesa.scss*

- **Mesa.html:**

Finalment, en el fitxer *html* es defineix la posició i quins elements es mostren en la pàgina de l'aplicació:

```

1  <ion-header>
2  <ion-navbar color="primary">
3    <button ion-button menuToggle>
4      <ion-icon name="menu"></ion-icon>
5    </button>
6    <ion-title>{{mesa.nombreAula}}</ion-title>
7  </ion-navbar>
8 </ion-header>
9 <ion-content>
10 <ion-row class="logo-row" *ngIf="mesa">
11   <ion-col></ion-col>
12   <ion-col width="57">
13     <img [src]="mesa.imageM" width="200px" height="200px" />
14   </ion-col>
15   <ion-col></ion-col>
16 </ion-row>
17 <ion-item>
18   {{ 'MESA.FILA' | translate }}
19   <ion-note item-right>{{ mesa.numFila }} </ion-note>
20 </ion-item>
21 <ion-item>
22   {{ 'MESA.COLUMNNA' | translate }}
23   <ion-note item-right>{{ mesa.numColumnna }} </ion-note>
24 </ion-item>
25 </ion-content>
26

```

Figura B.34: Nou arxiu *mesa.html*

És necessari saber que sempre que en aquest projecte trobeu un terme com ara “MESA.FILA” i “MESA.COLUMNNA”, seran **noms genèrics** creats per poder anomenar variables en les pàgines html de l'aplicació. D'aquesta manera, aquests noms genèrics seran declarats a l'arxiu **ca.json**, **es.json** o **en.json** (en el directori *src* \mapsto *assets* \mapsto *i18n*, tant del *mobile* com del *dashboard*) segons l'idioma en el qual els volguem expressar, juntament amb la seva correcta traducció a cadascun d'aquests idiomes (la qual es veurà ja directament a l'aplicació en comptes del nom genèric). Per tal d'especificar les variables de “MESA.TITLE”, “MESA.FILA” i “MESA.COLUMNNA”, s'han d'afegir en aquest cas particular al fitxer *es.json* al directori esmentat:

```

40  "MESA": {
41    "TITLE": "Mi mesa",
42    "FILa": "Fila",
43    "COLUMNNA": "Columnna"
44  },

```

Figura B.35: Etiquetes de traducció dels labels a l'arxiu *es.json*

Per últim, s'han de definir les pàgines creades en el fitxer **app.module.ts** ubicat al directori $src \mapsto app$. S'ha d'especificar la **MesaPage** dins de *la llista de les pàgines a l'inici* i a *declarations* i el servei de **MesaService** dins de *services*, *entryComponents* i *providers*.

B.3.6. Creació de les pàgines al repositori Dashboard

Tant el model com el servei són exactament iguals als de la part del Mobile, per tant es poden copiar directament les classes que ja hem creat prèviament al Mobile (a més a més dels models i els serveis que es troben al directori $src \mapsto app \mapsto shared$ dins de les seves respectives carpetes). També s'ha de definir la url en el fitxer **app.config.ts** que trobareu al directori $src \mapsto app$ tal com es va fer en el repositori del Mobile:

```
15 public static get GROUP_URL(): string { return this.SERVER_URL + '/api/groups'; }
16 public static get MESA_URL(): string { return this.SERVER_URL + '/api/mesas'; }
17
18 public static get LOGIN_URL(): string { return '/login'; }
19 public static get LOGOUT_URL(): string { return '/logout'; }
20 public static get MYSCHOOL_URL(): string { return '/school'; }
21 public static get TEACHERS_URL(): string { return '/teachers'; }
22 public static get STUDENTS_URL(): string { return '/students'; }
23 public static get COUNT_URL(): string { return '/count'; }
24 public static get GROUPS_URL(): string { return '/groups'; }
25 public static get MYMESA_URL(): string { return '/mesa'; }
```

Figura B.36: Definició de les URLs per al Dashboard

La única diferència és que s'ha d'afegir l'exportació del servei de “Mesa” al fitxer **index.ts** al directori $src \mapsto app \mapsto shared \mapsto services$.

```
8 export * from './grade.service';
9 export * from './matter.service';
10 export * from './group.service';
11 export * from './mesa.service';
```

Figura B.37: Exportació del servei Mesa

Per crear la pàgina des d'on es mostrarà la informació de l'objecte “Mesa”, s'ha de crear de la mateixa manera que en el Mobile, un nou directori dins de la carpeta *pages* que contingui, com a mínim, els tres fitxers següents:

- **Mesa.ts:**

Primer s'ha de definir el component i la forma que tindrà la pàgina sencera (incloent els altres arxius):

```

12  @Component({
13      selector: 'app-mesa',
14      templateUrl: './mesa.html',
15      styleUrls: ['./mesa.scss']
16  })

```

Figura B.38: Definició del component a l'arxiu *mesa.ts*

Després, a diferència de la part Mobile, la petició es fa directament a la pàgina de “Mesa” a dins del mètode que s'executa en quan s'obre la pàgina (**ngOnInit()**), ja que només hi ha un botó que porti a la pàgina.

Al Dashboard, el que abans anomenàvem pàgines al Mobile (per exemple, **Mesa-Page**) ara s'anomenaran components (**MesaComponent**).

```

27  export class MesaComponent implements OnInit {
28      public mesas: Mesa;
29      public myRole: Role;
30      public role = Role;
31
32      constructor(
33          public alertService: AlertService,
34          public utilsService: UtilsService,
35          public mesaService: MesaService,
36          public loadingService: LoadingService,
37          public snackbar: MatSnackBar) {
38
39          this.utilsService.currentUser = login.toObject(localStorage.getItem(AppConfig.IS_USER));
40          this.utilsService.role = Number(localStorage.getItem(AppConfig.IS_ROLE));
41          this.myRole = this.utilsService.role;
42      }
43
44      ngOnInit(): void {
45          if (this.myRole === Role.STUDENT) {
46              this.loadingService.show();
47              this.mesaService.getMyMesa().subscribe(
48                  (mesa: Mesa) => {
49                      this.loadingService.hide();
50                      this.mesas = mesa;
51                  },
52                  (error: Response) => {
53                      this.loadingService.hide();
54                      this.alertService.show(error.toString());
55                  }
56              );
57          }
58      }
59  }

```

Figura B.39: Resta del nou arxiu *mesa.ts*

- **Mesa.scss:**

En aquest fitxer simplement tornem a definir l'estil de la pàgina:

```
1  .mesa-content {
2      margin: 40px auto;
3      max-width: 400px;
4  }
5
6  mat-form-field {
7      width: 100%;
8  }
```

Figura B.40: Nou arxiu *mesa.scss*

- **Mesa.html:**

Finalment el fitxer html defineix com està estructurada la pàgina la qual, en el nostre cas, només es tracta de la imatge a la part de dalt i dues línies d'informació:

```
1  <div class="mesa-content">
2      <h3 *ngIf="myRole == role.STUDENT">{{ 'MESA.TITLE' | translate }}</h3>
3      <mat-card *ngIf="mesas">
4          <mat-card-header>
5              <img [src]="mesas.imageM">
6              <mat-card-title>{{ mesas.nombreAula }}</mat-card-title>
7              <mat-card-subtitle>{{ 'Columna: ' }} {{ mesas.numColumna }}</mat-card-subtitle>
8              <mat-card-subtitle>{{ 'Fila: ' }} {{ mesas.numFila }}</mat-card-subtitle>
9          </mat-card-header>
10     </mat-card>
11 </div>
```

Figura B.41: Nou arxiu *mesa.html*

Per tal de poder referenciar la pàgina, s'han d'afegir les següents línies de codi en els fitxers especificats:

- Fitxer **navbar.html**, al directori *src* \mapsto *app* \mapsto *shared* \mapsto *navbar*:

Aquests tres fitxers *navBar* corresponen a la barra de navegació de la part superior de la pàgina on hi podem trobar tots els botons.

Per crear el botó de “Mesa” simplement s'ha d'afegir una línia abans de la corresponent al botó de *logout* que tingui com a **routerLink** el direccionament a la pàgina de mesa (juntament amb el títol que es vol mostrar a dins del botó):

```
6 <a mat-button class="docs-navbar-hide-small docs-button" routerLink="/home">{{ 'HOME.TITLE' | translate }}</a>
7 <a mat-button class="docs-navbar-hide-small docs-button" routerLink="/groups">{{ 'GROUPS.TITLE' | translate }}</a>
8 <a mat-button class="docs-navbar-hide-small docs-button" routerLink="/mesa">{{ 'MESA.TITLE' | translate }}</a>
9 <a mat-button class="docs-navbar-hide-small docs-button" routerLink="/login">{{ 'APP.LOGOUT' | translate }}</a>
```

Figura B.42: Botó de direccionament a la pàgina Mesa

- Fitxer **app.module.ts** dins la carpeta *src* \mapsto *app*:

Tal com s'ha fet en la part del Mobile, s'ha d'afegir la declaració *MesaComponent* en la part de *declarations* i *MesaService* dins la part de *providers*.

- Fitxer **app.routing.ts** dins la carpeta *src* \mapsto *app*:

Per últim, s'ha d'afegir el direccionament a la pàgina afegint una línia amb el *path* i el component al qual es dirigeix:

```
11 const appRoutes: Routes = [
12
13   // authenticated pages
14   { path: 'home', component: HomeComponent, canActivate: [AuthGuard] },
15   { path: 'groups', component: GroupsComponent, canActivate: [AuthGuard] },
16   { path: 'mesa', component: MesaComponent, canActivate: [AuthGuard] },
17   { path: '', component: HomeComponent, canActivate: [AuthGuard] },
18
19   // unauthenticated pages
20   { path: 'login', component: LoginComponent },
```

Figura B.43: Definició del path de la pàgina Mesa

APÈNDIX C. MANUAL DE GIT

En aquest apèndix es vol explicar la metodologia de treball a seguir en cas que en el projecte de l'aplicació Classpip s'estigui treballant amb diferents branques simultàniament o en cas que una de les branques estigui sent desenvolupada per un equip de diverses persones. Serà imprescindible familiaritzar-se amb el conjunt d'instruccions necessàries per treballar amb el software **GitHub**, el qual és una poderosa eina de treball en equip quan es desenvolupa una aplicació basada en codi de programació.

C.1. Primers passos amb el repositori Classpip

Per començar a treballar amb un dels projectes, n'hi ha prou amb clonar el projecte en el nostre equip, és a dir, fer una còpia en local que podrem modificar des del nostre ordinador i no afectarà a la branca del projecte que estigui penjat a la web del GitHub. Per fer això n'hi haurà prou amb obrir el terminal (el **cmd** a Windows), anar a la carpeta on es vulguin tenir els projectes (o obrir el terminal directament a la carpeta fent click dret amb el ratolí) i utilitzar la següent comanda:

```
git clone <url-del-repositori>
```

Una vegada s'ha clonat el projecte en el nostre equip s'entra en la seva carpeta corresponent, i des d'allà es pot veure l'estat (*git status*), les branques del projecte (*git branch*) i les seves versions (*git tag*). Un cop baixat, l'estat es mostrarà sense canvis, i només es veurà la branca **master**.

Els repositoris tenen la branca master, la dev, la test i la tutorial (les funcionalitats de cada una s'expliquen a la secció següent). Per obtenir una branca concreta del repositori és tan senzill com utilitzar les següents ordres:

```
git fetch origin <nom-branca>
git checkout <nom-branca>
```

Després d'això, tot el codi del projecte canviarà per mostrar el codi que hi ha a la branca tutorial, on es veuran els fitxers afegits en el tutorial de desenvolupament per Classpip.

Quan es canviï d'una versió a una altra dins del repositori, és bastant probable que els mòduls de node instal·lats hagin canviat de versió, no s'hagin instal·lat en aquesta versió, o s'hagin instal·lat mòduls nous que no estaven a la versió anterior. Per això, és recomanable esborrar la carpeta node_modules i tornar a realitzar la instal·lació de mòduls:

```
rm -rf node_modules
npm install
```

C.2. Protocol per treballar en grup en Git

La utilització d'un programa de control de versions, com Git en aquest cas, suposa un gran avenç a l'hora de realitzar treball en equip en qualsevol programari.

És convenient marcar unes pautes de treball en equip per facilitar la coordinació de grups de treball en el mateix projecte, agilitzant d'aquesta manera el treball de diverses característiques o mòduls de forma simultània i la seva posterior integració en el projecte.

Es parteix de la premissa que actualment hi ha dues branques, la master i la dev. Tenint en compte això, cada projecte nou començarà amb una nova branca que sorgirà de la versió actual de màster. L'objectiu és tenir les branques màster i dev sense diferències entre elles per poder començar a treballar òptimament. Per tant, es crearà una branca extra idèntica a màster (una còpia en el nostre entorn local abans de començar a modificar el projecte) on es realitzaran les modificacions pertinents a partir de la base ja esmentada. Un cop provades les millores i després d'haver vist que són funcionals, s'hauran de pujar els canvis adequadament a les branques de dev i, més tard, a master tal i com s'explica més endavant. Cada projecte tindrà un nom únic i identificatiu per a la seva branca. A partir d'ara ja es pot considerar:

- La branca master com la branca de producció, és a dir, la que es veurà públicament i utilitzaran els usuaris.
- La branca dev com la branca de preproducció, una branca que serà idèntica a la de producció per provar els desenvolupaments abans que siguin publicats en producció.
- La branca pròpia de cada projecte com la branca on cada estudiant o grup d'estudiants anirà integrant els diversos desenvolupaments del seu projecte.

El primer és fixar un temps de desenvolupament, unes funcionalitats a desenvolupar o ambdues coses simultàniament seguint una metodologia scrum. Un cop passat el temps fixat, es publicaran els desenvolupaments que s'hagin realitzat correctament durant aquest període de temps. En el cas d'haver fixat funcionalitats en lloc de temps, es publicaran les funcionalitats un cop acabades. L'elecció d'un mètode o un altre haurà de ser escollida per l'equip de treball.

El següent pas és clonar el projecte, i preparar-lo amb les branques master i test. Quan ja estan les dues branques llestes, tan sols cal crear una nova branca per al projecte. És important estar situat a la branca master quan es crea una branca nova perquè aquesta sorgeixi de master. La creació d'aquesta branca es fa mitjançant la comanda:

```
git checkout master  
git checkout -b <nom-nova-branca>
```

Una vegada que es té preparat l'entorn de treball amb les tres branques en la mateixa situació, es podrà començar una metodologia de treball en equip que faciliti la coordinació entre els membres de l'equip i que asseguri el correcte testeig dels treballs individuals de cada un.

Per començar un nou desenvolupament s'haurà de crear una nova branca a partir de la

branca master, que anomenarem a mode d'exemple 'desenvolupament1'. Això es realitza amb les instruccions següents:

```
git checkout master
git checkout -b desenvolupament1
```

En aquest moment la branca desenvolupament1 és exactament igual que la branca master, dev i la del projecte. I aquí és on es comença a desenvolupar la nova funcionalitat que es vulgui afegir al projecte.

La comanda que ens permetrà confirmar canvis realitzats sobre la branca en la que estem treballant s'anomena '*commit*'. No s'hauria de fer cap commit fins que aquest nou desenvolupament no estigui finalitzat i sigui funcional. Si es realitzen molts commits, en el cas d'haver de realitzar una revisió de la feina per trobar un error o per millorar alguna cosa, caldrà revisar els canvis realitzats en cadascun d'aquests individualment, cosa que dificulta bastant la revisió. Tenir tot el treball realitzat en un sol commit facilitarà la posterior revisió dels canvis realitzats, de cara a tenir-los tots junts en una mateixa versió.

Un cop finalitzat el nou desenvolupament, es realitzarà un '*merge*' contra la branca del projecte, ordre que ens permetrà fusionar la nostra branca amb la de preproducció i més tard contra la de producció (un cop n'estem segurs). Es realitzaran les proves pertinents per comprovar el correcte funcionament del projecte.

Per realitzar el merge sobre la branca del projecte (anomenada a partir d'ara projecte a mode d'exemple), es farà de la següent manera:

1. `git checkout projecte`
2. `git pull origin projecte`
3. `merge --no-ff --no-commit desenvolupament1`
4. `git commit -m "[projecte] merge with desenvolupament1"`

Cal destacar, que abans de realitzar el merge, és important fer un '*pull*' com s'indica en el pas 2, en el qual es descarregarien tots els possibles canvis a la branca realitzats per un altre membre del grup, i amb el que es realitzaria el merge sobre una branca actualitzada. És a dir, la instrucció pull ens permet descarregar al nostre entorn local el que hi ha a la branca demanada amb el pull.

Quan el desenvolupament es troba correctament finalitzat i és completament funcional en projecte, es realitza un merge sobre dev. El procés de merge a dev és el mateix que en projecte:

1. `git checkout dev`
2. `git pull origin dev`
3. `merge --no-ff --no-commit desenvolupament1`
4. `git commit -m "[dev] merge with desenvolupament1"`

En cas d'haver un conflicte entre els passos 3 i 4, aquest no quedarà commitejat gràcies al flag `--no-commit`, llavors n'hi haurà prou amb solucionar el conflicte, afegir els canvis amb un `git add` i commitejar segons el pas 4:

1. Solucionar conflicte
2. `git add --A`
3. `git commit --m "[dev] merge with desenvolupament1"`

En la figura C.1 queda resumida la metodologia per treballar en equip amb el Git.

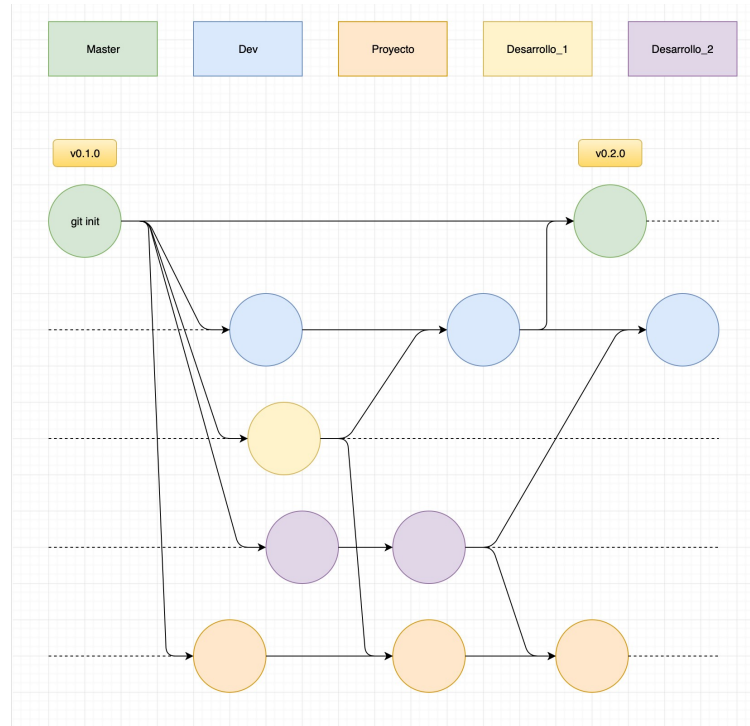


Figura C.1: Exemple de línia de treball

C.3. Exemple de treball amb el Git: Posada en marxa de la darrera versió de Classpip

En aquesta darrera secció del capítol de Git de l'Apèndix es pretén explicar els passos a seguir per tal de poder utilitzar de forma adequada la versió de Classpip desenvolupada en aquest projecte. D'aquesta manera, si nous membres del projecte volen testejar la interfície o fer-hi proves per tractar d'enfocar les noves millores del seu projecte, hauran de seguir aquest petit tutorial per tal de fer-ho. Així mateix, també servirà a professors que vulguin provar-la i començar-hi a jugar, donat que aquesta versió ja presenta una aplicació vendible amb unes funcionalitats pròpies d'una eina per a la gamificació. Els passos a seguir seran els següents:

1. Crear una carpeta buida on hi volguem clonar (guardar) els repositoris de Classpip per poder treballar amb ells.
2. Obrir la finestra d'instruccions (cmd) a dins de la carpeta creada i executar la següent ordre: *git clone **repositori***, on repositori ha de ser substituït per la url d'origen del repositori, on es troba emmagatzemat (cal executar l'ordre una vegada per a cada repositori). Les url són les següents:
 - Dashboard: <https://github.com/alejandromartincruz/classpip-dashboard>
 - Services: <https://github.com/alejandromartincruz/classpip-services>
3. Instal·lar els repositoris executant la finestra de cmd a dins de la carpeta de cada repositori, fent servir la següent instrucció: *npm install*.
4. Fent servir el cmd a dins de la carpeta de cada repositori, caldrà executar la següent seqüència d'ordres per tal d'obtenir la branca més actualitzada:
 - En executar *git status* serem informats de que som a la branca **master** (la d'origen, sense canvis) i que en realitzar la instal·lació s'ha modificat un arxiu: **modified: package-lock.json**. Llavors hauréu d'executar la comanda *git add -A* i més tard *git commit -m 'cambio'* per tal que el Git ens permeti canviar de branca.
 - A continuació executarem *git checkout -b Exemple*, el qual ens crearà una nova branca anomenada 'Exemple' i ens posicionarà en ella de manera directa, on farem el pull. Si fem *git branch* veurem que existeixen dues branques: master i Exemple, sent la darrera on ens trobem actualment.
 - Executem l'ordre *git pull origin **branca*** per tal de fer merge amb la branca amb la que volguem treballar, on branca haurà de substituir-se pel nom de la branca que volguem obtenir. En el nostre cas la branca més actualitzada de Dashboard s'anomena **VersionDefinitiva**, mentre que la de Services es diu **VersionFinal**.
5. Cal executar la instrucció *node .* en el repositori de Services per tal d'iniciar-lo (dins la finestra cmd oberta a la carpeta del Services).

6. Per últim, fer servir la comanda *npm run start* al cmd del repositori Dashboard i, un cop hagi compilat, només caldrà obrir la url del Dashboard per tal de començar a jugar. Les urls per accedir tant a l'API del Services com a la pròpia aplicació de Classpip en versió Dashboard són les següents:

- Dashboard: <http://localhost:4200/>
- Services: <http://localhost:3000/explorer/>

APÈNDIX D. CATÀLEG D'ERRORS

D.1. Errors a Windows

D.1.1. Error 1: Can't find Python executable:

```
C:\Users\dcastella\AppData\Roaming\npm\node_modules\strongloop\node_modules\utf-8-validate>if not defined np
m_config_node_gyp (node "C:\Program Files\nodejs\node_modules\npm\bin\node-gyp-bin\..\..\node_modules\node-
gyp\bin\node-gyp.js" rebuild ) else (node "" rebuild )
gyp ERR! configure error
gyp ERR! stack Error: Can't find Python executable "python", you can set the PYTHON env variable.
gyp ERR! stack     at PythonFinder.failNoPython (C:\Program Files\nodejs\node_modules\npm\node_modules\node-
gyp\lib\configure.js:483:19)
gyp ERR! stack     at PythonFinder.<anonymous> (C:\Program Files\nodejs\node_modules\npm\node_modules\node-
gyp\lib\configure.js:508:16)
gyp ERR! stack     at C:\Program Files\nodejs\node_modules\npm\node_modules\graceful-fs\polyfills.js:284:29
gyp ERR! stack     at FSReqWrap.oncomplete (fs.js:153:21)
gyp ERR! System Windows_NT 10.0.16299
gyp ERR! command "C:\\Program Files\\nodejs\\node.exe" "C:\\Program Files\\nodejs\\node_modules\\npm\\node_m
odules\\node-gyp\\bin\\node-gyp.js" "rebuild"
gyp ERR! cwd C:\Users\dcastella\AppData\Roaming\npm\node_modules\strongloop\node_modules\utf-8-validate
gyp ERR! node -v v8.6.0
gyp ERR! node-gyp -v v3.6.2
gyp ERR! not ok

> bufferutil@1.2.1 install C:\Users\dcastella\AppData\Roaming\npm\node_modules\strongloop\node_modules\ws\no
de_modules\bufferutil
> node-gyp rebuild
```

Figura D.1: Error can't find Python executable

L'error surt per primera vegada en instal·lar l'*strongloop* en el mòdul de services, però ja que es tracta d'una eina, a l'hora d'executar els serveis no dóna cap problema.

L'error torna a sortir en fer el set up del mòdul de Mobile quan executem la instrucció "*npm install*":

```

ments\\classpip-mobile\\node_modules\\which\\which.js:68:19)\n    at E (C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\which\\which.js:80:29)\n    at C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\which\\which.js:89:16\n    at C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\isexe\\index.js:42:5\n    at C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\isexe\\windows.js:36:5\n    at FSReqWrap.oncomplete (fs.js:153:21)',\ngyp verb 'which' failed code: 'ENOENT' )\ngyp verb could not find "python". checking python launcher\ngyp verb could not find "python". guessing location\ngyp verb ensuring that file exists: C:\\Python27\\python.exe\ngyp ERR! configure error\ngyp ERR! stack Error: Can't find Python executable "python", you can set the PYTHON env variable.\ngyp ERR! stack at PythonFinder.failNoPython (C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\node-gyp\\lib\\configure.js:483:19)\ngyp ERR! stack at PythonFinder.<anonymous> (C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\node-gyp\\lib\\configure.js:508:16)\ngyp ERR! stack at C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\graceful-fs\\polyfills.js:284:29\ngyp ERR! stack at FSReqWrap.oncomplete (fs.js:153:21)\ngyp ERR! System Windows_NT 10.0.16299\ngyp ERR! command "C:\\Program Files\\nodejs\\node.exe" "C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\node-gyp\\bin\\node-gyp.js" "rebuild" "--verbose" "--libsass_ext=" "--libsass_cflags=" "--libsass_ldflags=" "--libsass_library=" \ngyp ERR! cwd C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\node-sass\ngyp ERR! node -v v8.6.0\ngyp ERR! node-gyp -v v3.6.2\ngyp ERR! not ok\nBuild failed\nnpm WARN Error: EPERM: operation not permitted, scandir 'C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\are-we-there-yet\\node_modules'\nnpm WARN { Error: EPERM: operation not permitted, scandir 'C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\are-we-there-yet\\node_modules'\nnpm WARN stack: 'Error: EPERM: operation not permitted, scandir 'C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\are-we-there-yet\\node_modules'',\nnpm WARN errno: -4048,\nnpm WARN code: 'EPERM',\nnpm WARN syscall: 'scandir',\nnpm WARN path: 'C:\\Users\\dcastella\\Documents\\classpip-mobile\\node_modules\\are-we-there-yet\\node_modules' }\nnpm ERR! code ELIFECYCLE\nnpm ERR! errno 1\nnpm ERR! node-sass@3.10.1 postinstall: `node scripts/build.js`\nnpm ERR! Exit status 1\nnpm ERR!\nnpm ERR! Failed at the node-sass@3.10.1 postinstall script.\nnpm ERR! This is probably not a problem with npm. There is likely additional logging output above.\n\nnpm ERR! A complete log of this run can be found in:\nnpm ERR! C:\\Users\\dcastella\\AppData\\Roaming\\npm-cache\\_logs\\2018-05-13T21_41_53_483Z-debug.log

```

Figura D.2: Error can't find Python executable

Simplement executant la següent instrucció al terminal s'instal·laran tots els components necessaris per dur a terme el set up sense errors:

```
npm install --global --production windows-build-tools
```

D.1.2. Error 2: Unable to clone Git repository due to self signed certificate:

En intentar clonar un repositori del GIT en el nostre entorn de treball local és possible que es doni el següent missatge d'error:

```
2013-07-02 18:15:48,400 WARN [InitialPinger2 ] fisheye IndexingPingRequest-doRequest
- Exception during FishEye Incremental Indexing of Gitrepo: com.cenqua.fisheye.config.ConfigException:
Unable to clone remote repository: https://fisheyeuser@git.server:8443/scm/test/gitrepo.git
- [fatal: unable to access 'https://fisheyeuser:XXXX@git.server:8443/scm/test/gitrepo.git':
SSL certificate problem: self signed certificate in certificate chain] - Cloning into bare re-
pository 'clone'...
```

Es soluciona deshabilitant la verificació del GIT SSL en el server hosting file amb la següent instrucció:

```
git config --global http.sslVerify false
```

D.1.3. Error 3: Error de connexió amb la base de dades:

Quan es treballi amb **versions més antigues** (com per exemple amb la d'en Ferran o la de l'Anna i en Joan, on aquest error encara no es trobava corregit) per tal de realitzar els tutorials d'aprenentatge, pot sorgir un error durant la connexió amb la base de dades. En versions més avançades aquest error ja es troba corregit, però en casos particulars d'ús de versions antigues pot seguir donant problemes. Si al terminal surt un error anomenant la base de dades '**dB**', cal que aneu al directori del *services* \mapsto *server* i al fitxer **datasources.json** comproveu que hi són exactament les següents línies (si no hi són canvieu el que hi hagi pel que es mostra a la figura següent):

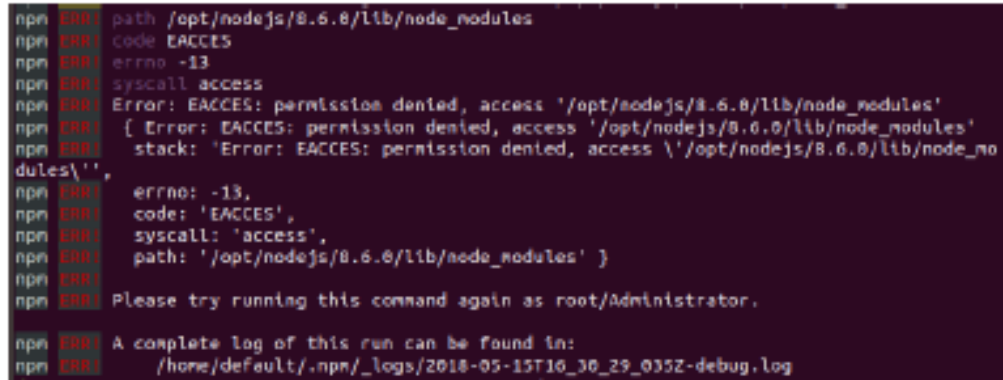
```
2      "db": {
3          "name": "db",
4          "connector": "memory"
5      },
```

Figura D.3: Connexió amb la dB a *datasources.json*

D'aquesta manera es podrà treballar amb una base de dades en local i l'aplicació no intentarà connectar amb una externa que es trobarà desactivada o no funcional.

D.2. Errors a Ubuntu

D.2.1. Error 1: Permission denied:



```
npm ERR! path /opt/nodejs/8.6.0/lib/node_modules
npm ERR! code EACCES
npm ERR! errno -13
npm ERR! syscall access
npm ERR! Error: EACCES: permission denied, access '/opt/nodejs/8.6.0/lib/node_modules'
npm ERR! { Error: EACCES: permission denied, access '/opt/nodejs/8.6.0/lib/node_modules'
npm ERR!   stack: 'Error: EACCES: permission denied, access \'/opt/nodejs/8.6.0/lib/node_no
npm ERR! dules\'',
npm ERR!   errno: -13,
npm ERR!   code: 'EACCES',
npm ERR!   syscall: 'access',
npm ERR!   path: '/opt/nodejs/8.6.0/lib/node_modules' }
npm ERR! Please try running this command again as root/Administrator.

npm ERR! A complete log of this run can be found in:
npm ERR!   /home/default/.npm/_logs/2018-05-15T16_36_29_035Z-debug.log
```

Figura D.4: Error permission denied

Aquest error apareix quan no s'executa una instrucció amb permisos d'administració. Es soluciona posant “**sudo**” al davant de la instrucció.

D.2.2. Error 2: Bucle de l'StrongLoop:

```
gyp WARN! DACCES attempting to reinstall using temporary dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp"
gyp WARN! DACCES user "root" does not have permission to access the dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp/8.6.0"
gyp WARN! DACCES attempting to reinstall using temporary dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp"
gyp WARN! DACCES user "root" does not have permission to access the dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp/8.6.0"
gyp WARN! DACCES attempting to reinstall using temporary dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp"
gyp WARN! DACCES user "root" does not have permission to access the dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp/8.6.0"
gyp WARN! DACCES attempting to reinstall using temporary dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp"
gyp WARN! DACCES user "root" does not have permission to access the dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp/8.6.0"
gyp WARN! DACCES attempting to reinstall using temporary dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp"
gyp WARN! DACCES user "root" does not have permission to access the dev dir "/opt/nodejs/8.6.0/lib/node_modules/strongloop/node_modules/heapdump/.node-gyp/8.6.0"
```

Figura D.5: Error del bucle de l'StrongLoop

En instal·lar l'StrongLoop de la part del services, a la part final de la instal·lació s'entra en bucle que es pot aturar mitjançant la comanda *Crtl+C*. No s'ha observat cap implicació en continuar la instal·lació.

D.2.3. Error 3: Not found: make:

```

gyp verb node dev dir /home/default/.node-gyp/0.6.0
gyp verb build error
gyp verb stack Error: not found: make
gyp verb stack   at getNotFoundError (/home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:11:12)
gyp verb stack   at F (/home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:48:19)
gyp verb stack   at E (/home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:50:19)
gyp verb stack   at /home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:86:16
gyp verb stack   at /home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:42:15
gyp verb stack   at /home/default/.Documents/repos/classip-mobile/node_modules/which/which.js:86:16
gyp verb stack   at FSReqWrap.oncomplete (fs.js:153:21)
gyp verb System Linux 4.13.0-39-generic
gyp verb command "opt/node.js/0.6.0/bin/node" /home/default/.Documents/repos/classip-mobile/node_modules/node-gyp/bin/node-gyp.js "rebuild" "--verbose" "--libsass_ext=" "--libsass_flags=" "--libsass_defflags=" "--libsass_library="
gyp verb cwd /home/default/.Documents/repos/classip-mobile/node_modules/node-gyp
gyp verb node -v v0.6.0
gyp verb node-gyp -v v0.6.2
gyp verb not ok
gyp verb build failed
npm verb optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules/fsevents):
npm verb optional SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: want {"os": "darwin", "arch": "any"} (current: {"os": "linux", "arch": "x64"})
npm verb code ELIFECYCLE
npm verb error 1
npm verb node-sass@3.18.1 postinstall: "node scripts/build.js"
npm verb exit status 1
npm verb
npm verb failed at the node-sass@3.18.1 postinstall script.
npm verb this is probably not a problem with npm, there is likely additional logging output above.

npm verb A complete log of this run can be found in:
npm verb /home/default/.npm/_logs/2016-05-15T10_37_33_997Z-debug.log
default@ip-172-31-17-10:~$ cd /home/default/.Documents/repos/classip-mobile

```

Figura D.6: Error not found: make

Aquest error pot aparèixer en executar la instrucció `npm install`, després d'instal·lar l'ionic i el cordova en la part del Mobile. Es pot solucionar executant la instrucció:

```
sudo apt-get install build-essential
```

Es tracta d'una comanda que permet instal·lar una llista de paquets necessaris per compilar en C/C++.

D.2.4. Error 4: Plugins no trobats:

En executar la instrucció “*cordova prepare*” apareixen una sèrie de warnings de plugins que no s’han trobat:

[illegible]

Figura D.7: Informació dels plugins no trobats

Es mostra un error al final que diu que no troba l'entorn de "ANDROID-HOME". S'hauran d'executar les següents instruccions per solucionar-ho:

```
export ANDROID_HOME=/usr/local/android-sdk-linux/tools
export PATH="$PATH:/usr/local/android-sdk-linux/platform-tools"
```

En executar-les es defineix l'entorn automàticament i en introduir la instrucció “*cordova prepare*”, es redueix la llista de plugins a un:

```
default@tty-OptiFlex-7910:~/Documents/repes/classpfp-mobile$ export ANDROID_HOME=/usr/local/
android-sdk-linux/tools
default@tty-OptiFlex-7910:~/Documents/repes/classpfp-mobile$ export PATH=$PATH:/usr/local/
android-sdk-linux/platform-tools
default@tty-OptiFlex-7910:~/Documents/repes/classpfp-mobile$ cordova prepare
discovered plugin "ms.fernerya.callnumber" in config.xml. Adding it to the project
Fetching plugin "ms.fernerya.callnumber@0.0.2" via npm
Installing "call-number" for android
Failed to install 'call-number':Error
at /home/default/Documents/repes/classpfp-mobile/platforms/android/cordova/lib/check_r
eq.js:226:19
at _fulfilled (/home/default/Documents/repes/classpfp-mobile/platforms/android/cordova
/node_modules/q/q.js:834:14)
at self.promiseDispatch.done (/home/default/Documents/repes/classpfp-mobile/platforms/
android/cordova/node_modules/q/q.js:883:18)
at Promise.prototype.promiseDispatch (/home/default/Documents/repes/classpfp-mobile/plat
forms/android/cordova/node_modules/q/q.js:796:12)
at /home/default/Documents/repes/classpfp-mobile/platforms/android/cordova/node_modul
e/q/q.js:857:14
at runSingle (/home/default/Documents/repes/classpfp-mobile/platforms/android/cordova/
node_modules/q/q.js:117:11)
at flush (/home/default/Documents/repes/classpfp-mobile/platforms/android/cordova/node
_modules/q/q.js:125:13)
at _combinedTickCallback (internal/process/next_tick.js:131:7)
at process._tickCallback (internal/process/next_tick.js:166:9)
Failed to restore plugin "ms.fernerya.callnumber" from config.xml. You might need to try ad
ding it again. Error: 'ANDROID_HOME' environment variable is set to non-existent path: /usr
/local/android-sdk-linux/tools
Try update it manually to point to valid SDK directory.
default@tty-OptiFlex-7910:~/Documents/repes/classpfp-mobile$
```

Figura D.8: Plugin que es queda sense identificar

D.3. Errors generals

D.3.1. Error 1: Error Visual Package 8.1 SDK:

Quan es fa la instal·lació del **Visual Studio Code**, si surt un error en el qual s'especifica que s'ha d'instal·lar el SDK 8.1 i ja es té instal·lat, s'ha d'anar a: *Panell de control* → *Programes* → *Programes i Característiques*:

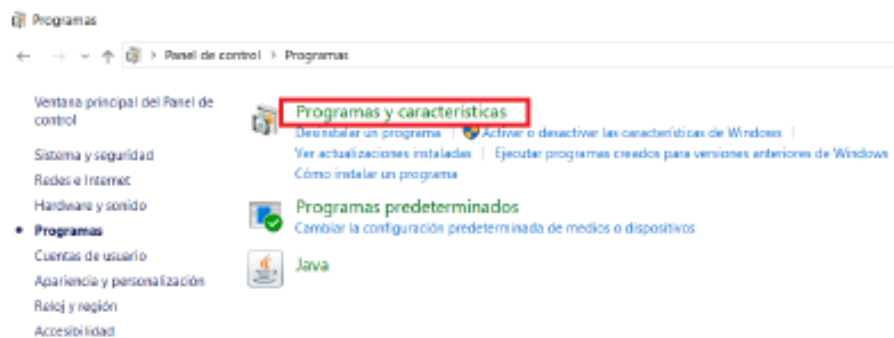
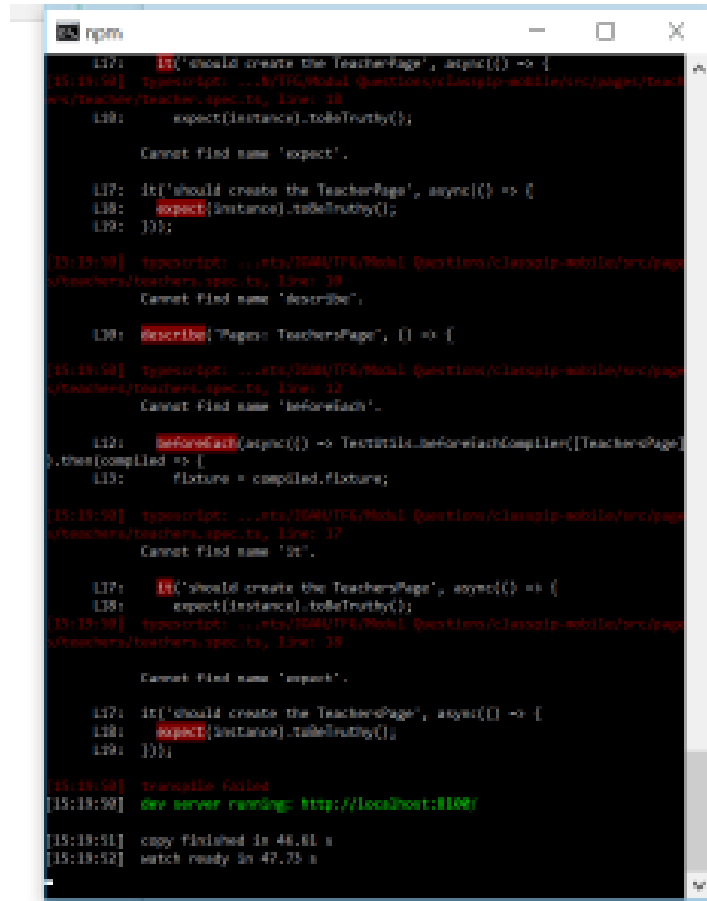


Figura D.9: Panell de control de l'ordinador

En la llista de programes, es selecciona el Visual Studio Code, es fa click a botó dret i es selecciona 'Canviar'. En el formulari que s'obre, seleccionar l'SDK 8.1 per a descarregar-ho.

D.3.2. Error 2: Error Visual Package 8.1 SDK:

Al moment de fer la instal·lació del repositori del Mobile, si quan executem la instrucció “cordova prepare” surt el següent error:



```
npm
117: it('should create the TeacherPage', async() => {
[15:18:58] typescript: ...src/IDE4/TP6/Model Questions/classip-mobile/src/page
s/teachers/teachers.spec.ts, line: 18
118:   expect(instance).toBeTruthy();
119: });
    Cannot find name 'expect'.

117: it('should create the TeacherPage', async() => {
118:   expect(instance).toBeTruthy();
119: });
[15:18:58] typescript: ...src/IDE4/TP6/Model Questions/classip-mobile/src/page
s/teachers/teachers.spec.ts, line: 18
    Cannot find name 'describe'.

118: describe('Pages: TeachersPage', () => {
[15:18:58] typescript: ...src/IDE4/TP6/Model Questions/classip-mobile/src/page
s/teachers/teachers.spec.ts, line: 12
    Cannot find name 'beforeEach'.

112: beforeEach(async() => TestUtils.beforeEachCompiler([TeacherPage]
).then(compiled => {
113:   fixture = compiled.fixture;
[15:18:58] typescript: ...src/IDE4/TP6/Model Questions/classip-mobile/src/page
s/teachers/teachers.spec.ts, line: 17
    Cannot find name 'it'.

117: it('should create the TeachersPage', async() => {
118:   expect(instance).toBeTruthy();
[15:18:58] typescript: ...src/IDE4/TP6/Model Questions/classip-mobile/src/page
s/teachers/teachers.spec.ts, line: 18
    Cannot find name 'expect'.

117: it('should create the TeachersPage', async() => {
118:   expect(instance).toBeTruthy();
119: });
[15:18:58] transpile failed
[15:18:58] dev server running: http://localhost:8100/
[15:18:51] copy finished in 46.61 s
[15:18:52] watch ready in 47.73 s
```

Figura D.10: Error de Visual Package 8.1 SDK

S'haurà de solucionar afegint les següents línies en l'arxiu **tsconfig.json**, en la part “*exclude*” (el podreu trobar al directori general del repositori del Mobile):

```
"e2e",
"**/*spec.ts"
```

D.3.3. Error 3: Error intern del servidor:



Figura D.11: Missatge d'error intern en el servidor

Si en el moment d'executar l'aplicació i interactuar surt el missatge esmentat amunt, es pot deure a que el programa estigui apuntant al servidor de producció.

Si es vol desenvolupar, s'haurà de comentar i descomentar les següents línies del fitxer següent:

/classpip-mobile/src/app/app.config.ts

```
// public static get SERVER_URL(): string { return 'https://api.classpip.com'; } // PRO  
public static get SERVER_URL(): string { return 'http://localhost:3000'; } // DEV
```

D'aquesta manera, es podrà passar de fer una crida a un servidor extern que, a priori, es trobarà desactivat a treballar amb el nostre codi en local per tal de modificar i ampliar l'aplicació.